UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Škaper

SIMULACIJA OMREŽJA MPLS

DIPLOMSKO DELO NA
UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Nikolaj Zimic

Ljubljana, 2012

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01800/2012
Datum: 02.02.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGOR ŠKAPER**

Naslov: **SIMULACIJA OMREŽJA MPLS**
MPLS NETWORK SIMULATION

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Zaradi vse večjih zahtev uporabnikov po hitrem in zanesljivem prenosu podatkov, je bilo potrebno nadgraditi omrežja, ki uporabljajo protokol IP. Pomanjkljivost omrežij IP je predvsem v potratnem usmerjanju paketov ter nezmožnosti izvajanja prometnega inženiringa. Slabosti je moč omiliti z uvedbo protokola MPLS.

V diplomski nalogi izdelajte simulacijo omrežja, ki uporablja protokol MPLS. Rešitev primerjajte s funkcijsko enakovrednim omrežjem brez protokola MPLS. Pri simulaciji se osredotočite predvsem na možnost uporabe prometnega inženiringa ter vpliv le-tega na kvaliteto storitev. Rezultate simulacije ustrezno analizirajte ter posebej natančno obrazložite vse nepričakovane rezultate.

Mentor:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko
UNIVERZA V LJUBLJANI
2

Dekan:

prof. dr. Nikolaj Zimic

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Gregor Škaper,

z vpisno številko 63050112,

sem avtor diplomskega dela z naslovom:

SIMULACIJA OMREŽJA MPLS

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom

  prof. dr. Nikolaja Zimica

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 14.5.2012 Podpis avtorja:

Zahvala

Iskreno se zahvaljujem mentorju, prof. dr. Nikolaju Zimicu, za potrpežljivost, motivacijo, strokovno pomoč in vse napotke pri pisanju diplomskega dela.

Posebna zahvala je namenjena moji družini in partnerki, ki so mi skozi celoten študij stali ob strani in mi nudili podporo.

Zahvala gre tudi vsem ostalim, ki so na kakršen koli način pripomogli k nastanku diplomskega dela.

KAZALO

Kratice in simboli

ATM:	Asynchronous Transfer Mode
BA:	Behavior Aggregate
BB:	Backbone Router
BGP:	Border Gateway Protocol
BO:	Branch Office
CBR:	Constant Bit Rate
CoS:	Class of Service
CR-LDP:	Constraint-based Routing Label Distribution Protocol
DS:	Differentiated Services
DSCP:	Differentiated Services Code Point
EGP:	Exterior Gateway Protocol
ERB:	Explicit Routing Base
ER-LSP:	Explicitly Routed LSP
E-LSP:	EXP-inferred-PSC LSP
FEC:	Forwarding Equivalence Class
FIB:	Forwarding Information Base
FIFO:	First in, first out
FR:	Frame Relay
FTP:	File Transfer Protocol
HTTP:	Hyper-Text Transfer Protocol
IETF:	Internet Engineering Task Force
IGP:	Interior Gateway Protocol
IP:	Internet Protocol
IPSec:	IP Security
IPTV:	Internet Protocol TV
IPv4:	Internet Protocol Version 4

IPv6:	Internet Protocol Version 6
IS:	Integrated Services
LDP:	Label Distribution Protocol
LIB:	Label Information Base
LER:	Label Edge Router
LSP:	Label Switched Path
LSR:	Label Switch Router
L-LSP:	Label-only-inferred-PSC LSP
MPLS:	Multiprotocol Label Switching
MP-BGP:	Multi Protocol BGP
MTU:	Maximum Transmission Unit
OSI:	Open Systems Interconnection
OSPF:	Open Shortest Path First
PFT:	Partial Forwarding Table
PHB:	Per Hop Behaviour
PHP:	Penultimate Hop Popping
QoS:	Quality of Service
RIP:	Routing Information Protocol
RSVP:	Resource Reservation Protocol
SNMP:	Simple Network Management Protocol
TCP:	Transmission Control Protocol
ToS:	Type of Service
TTL:	Time-To-Live
UDP:	User Datagram Protocol
VoIP:	Voice over IP
VPN:	Virtual Private Network
VRF:	Virtual Route Forwarding

Povzetek

V diplomskem delu smo obravnavali protokol MPLS, ki je pomembna nadgradnja omrežja IP. Omrežje IP, ki ima implementiran protokol MPLS, se imenuje omrežje MPLS. Glavna ideja protokola MPLS je, da se v nepovezavno orientirano omrežje IP vpelje povezavno orientiran princip, to pomeni, da se pred prenosom IP-paketov vzpostavi povezava med končnima vozliščema. S tem se izboljšata hitrost in zanesljivost prenosa IP-paketov. Pomembna funkcionalnost protokola MPLS je prometni inženiring, s katerim se zagotovi preusmeritev prometnih tokov na želene poti v omrežju, to pa omogoča razbremenitev omrežja in optimalno izrabo omrežnih virov. V eksperimentalnem delu smo s simulacijo omrežja IP in omrežja MPLS preverili vpliv protokola MPLS na delovanje omrežja IP. Implementacija simulacijskih modelov in izvajanje simulacije sta potekala s pomočjo omrežnega simulatorja NS-2. Simulacijske modele smo implementirali v programskem jeziku OTcl. Največ pozornosti smo posvetili simulaciji prometnega inženiringa. Za omrežje IP in omrežje MPLS smo skozi simulacijo spremljali parametre kakovosti storitev, kot so zakasnitev, spremenljivost zakasnitve (jitter) in prepustnost. Simulacije smo izvajali za različne obremenitve omrežja in za različne vrste omrežnega prometa. Za pridobivanje rezultatov simulacij smo uporabili programski jezik AWK. Dobljeni rezultati simulacij so bili v skladu s pričakovanji, razen nekaterih zanimivih pojavov, ki pa smo jih ustrezno ovrednotili.

Ključne besede: omrežje IP, protokol MPLS, prometni inženiring, omrežni simulator NS-2, simulacija

Abstract

The thesis deals with the MPLS protocol, which represents an important upgrade of an IP network. An IP network which uses MPLS protocol is called an MPLS network. The main idea behind the MPLS protocol is the connection-oriented principle which is brought into a connectionless-oriented IP network, which means that a path is established between two end nodes before IP-packets are transferred. This improves the speed and reliability of IP-packet transmission. An important feature of MPLS protocol is traffic engineering. With traffic engineering the network traffic is redirected to the desired path in the network which provides less network load and optimal utilization of network resources. The experimental part of the study was aimed at simulating the IP network and MPLS network to check the impact of MPLS protocol on the operation of an IP network. The implementation of simulation models and the simulation process were performed using a network simulator NS-2. Simulation models were implemented with the OTcl programming language. In experimental work we focused on the simulation of traffic engineering. In the course of the simulation, we monitored certain quality of service parameters, e.g. delay, delay variation (jitter), and throughput. The simulation was carried out for different network loads and different types of network traffic. The simulation results, which were obtained using the AWK programming language, were in accordance with our expectations, except for a small number of interesting phenomena, which were properly evaluated.

Keywords: IP network, MPLS protocol, traffic engineering, network simulator NS-2, simulation

1 Uvod

Danes si težko predstavljamo življenje brez interneta. Na začetku je internet uporabnikom omogočal le izvajanje enostavnih aplikacij, kot so brskanje po spletnih straneh, pošiljanje elektronske pošte in podatkovnih datotek. Z razvojem interneta, vedno večjim porastom uporabnikov in naraščanjem njihovih potreb ter pričakovanj so se pojavile nove aplikacije, kot so VoIP (ang. Voice over IP), IPTV (ang. Internet Protocol TV) in multimedija, ki uporabnikom interneta poleg pregleda besedilnih vsebin omogočajo tudi ogled video in zvočnih vsebin. Z novimi aplikacijami se je povečala količina omrežnega prometa in pojavile so se stroge zahteve glede prenosnih parametrov. Pri tem se je tradicionalno omrežje IP, ki usmerja IP-pakete od izvora do ponora, izkazalo za neučinkovito. Z rastjo količine omrežnega prometa v omrežju IP so se začeli pojavljati problemi, povezani z zasičenjem omrežja, s predolgimi odzivnimi časi in z izgubami IP-paketov. Problemi se v omrežju IP pojavljajo predvsem zaradi načina prenosa, ker se IP-paketi med usmerjevalniki prenašajo nepovezavno (ang. conectionless). Vsak usmerjevalnik se mora na podlagi naslova v glavi IP-paketa sam odločati o nadaljnjem usmerjanju IP-paketov skozi omrežje IP. Slednje je izredno kompleksna in potratna procedura, ki za posredovanje velike količine omrežnega prometa porabi preveč časa in tako zmanjšuje zmogljivost omrežja. Če želimo povečati zmogljivost omrežja IP in tako omogočiti nemoteno delovanje aplikacij, je omrežje IP treba nadgraditi z novimi tehnologijami. Ena izmed takih tehnologij je protokol MPLS, ki se je v praksi že izkazal za zelo uporabnega in ga v polni meri uporabljajo ponudniki internetnih storitev. Protokol MPLS poleg hitrejšega prenosa podatkov omogoča tudi pomembno funkcionalnost, kot je prometni inženiring, s katerim se optimizira delovanje omrežja IP in poveča kakovost storitev za aplikacije, ki delujejo v realnem času.

Preden se je protokol MPLS popolnoma uveljavil v realnem omrežju IP, je bilo treba preveriti njegov vpliv na zmogljivost omrežja IP in preizkusiti funkcionalnosti, ki jih omogoča. V ta namen se uporabi simulacija, ki pomeni pomembno metodo na področju raziskav omrežij za preskušanje novih protokolov in sprememb obstoječih protokolov. Simulacija omrežja je zelo koristna za razumevanje delovanja omrežja ter je nepogrešljiv in izredno pomemben faktor pri nadgradnji telekomunikacijskih omrežij z novimi tehnologijami in protokoli. Simulacija igra ključno vlogo pri vrednotenju zmogljivosti omrežja v fazi načrtovanja, ker pridobimo na dveh pomembnih vidikih [30]. Prvi izmed vidikov je, da nam za preizkušanje lastnosti protokola in njegove zmogljivosti ni treba zasesti realnega omrežja. Drugi vidik se nanaša na uporabnike, saj slednji lahko opravljajo svoje naloge nemoteno in neodvisno od izvajanja simulacij. Tako lahko vzporedno poganjamo simulacije in pridobivamo pomembne podatke. V današnjem času sta testiranje in analiziranje velikih in kompleksnih omrežij v realnem okolju velik stroškovni zalogaj, zato se pogosto uporabljajo simulacijska orodja, s katerimi se dobi pomembne informacije o zmogljivosti omrežij. Predvsem na račun kompleksnosti omrežij se je v zadnjem času uporaba simulacijskih orodij bistveno povečala.

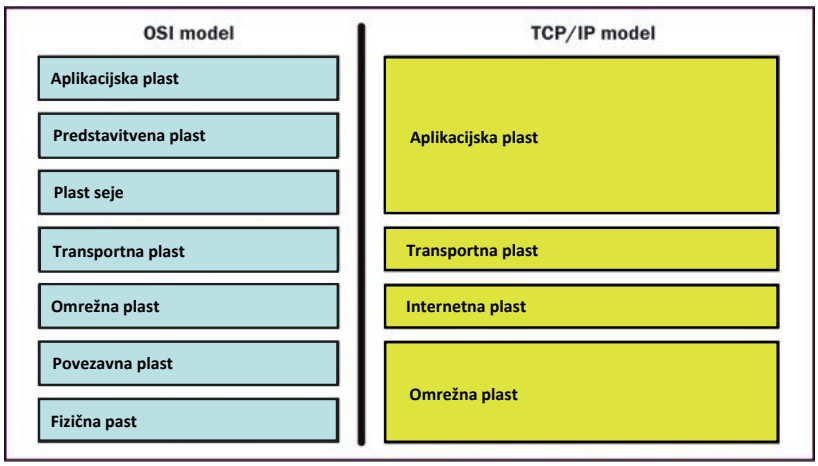
Cilj diplomske naloge je preveriti vpliv protokola MPLS na delovanje omrežja IP z izvajanjem simulacije s pomočjo omrežnega simulatorja NS-2. Pri tem se bomo osredotočili na simulacijo prometnega inženiringa, pri čemer bomo v omrežju MPLS vzpostavili eksplicitno pot. Simulacijo prometnega inženiringa bomo izvajali z različno obremenitvijo omrežja. Za omrežje IP in omrežje MPLS bomo skozi simulacijo spremljali parametre kakovosti storitev, ki so pomembni za kvaliteten prenos podatkov med aplikacijami in omrežji.

2 Računalniško omrežje

Računalniško omrežje lahko definiramo kot sistem med seboj povezanih neodvisnih naprav (računalniki, telefoni, tiskalniki itd.). V računalniško omrežje se lahko povezuje dve ali več naprav. Naloga omrežij je komunikacija in izmenjava podatkov med napravami. Za pravilno komuniciranje vseh naprav v omrežju morajo vse uporabljati enake komunikacijske protokole. Ti določajo pravila in način komunikacije med napravami. Komunikacijski protokol je strukturiran v obliki različnih dogovorov, postopkov in sporočil, ki vodijo in opravljajo prenos informacij v omrežju. Osnovna pravila za komunikacijo med omrežji predstavlja referenčni model OSI. Vsaka plast referenčnega modela OSI predstavlja različno raven komunikacije.

2.1 Referenčni model OSI

Referenčni model OSI je najbolj poznan reprezentativen model za izmenjavo podatkov v omrežju. Velja za osnovni arhitekturni model za omrežno komunikacijo med računalniki. Referenčni model OSI deli omrežno komunikacijo v sedem plasti. Vsaka plast pokriva različne omrežne funkcije. Konkretno so zgornje plasti modela OSI odgovorne za prenos podatkov k uporabniku, spodnje pa za povezovanje med omrežnimi elementi [11]. Poleg modela OSI poznamo še model TCP/IP, ki temelji na referenčnem modelu OSI in je sestavljen iz štirih plasti. Primerjavo nivojev teh dveh modelov prikazuje slika 1.



Slika 1: Primerjava plasti referenčnega modela OSI in TCP/IP [11]

2.1.1 Opis plasti v referenčnem modelu OSI

Referenčni model OSI je razdeljen na sedem plasti, kjer vsaka izmed plasti opravlja določeno funkcijo [9].

Fizična plast skrbi za dejanski prenos informacijskih signalov po komunikacijskem kanalu. Osnovna in najmanjša enota prenosa informacije je bit. Fizična plast definira električne in mehanske lastnosti mrežnih naprav. Določa tudi povezavo med omrežno napravo in prenosnim medijem (bakreni in optični kabel). Fizična plast skrbi za pretvorbo binarne informacije v električne signale.

Povezavna plast prenaša podatkovne okvire (ang. Frame) med dvema omrežnima napravama, ki sta povezani s prenosnim medijem. Osnovna naloga te plasti je zagotavljanje pravilnega prenašanja podatkov. Temeljne funkcije povezavne plasti so: razvrščanje bitov v podatkovne okvire, sinhronizacija okvirov in zaznavanje ter odpravljanje napak, ki se pojavijo pri prenosu podatkov. Danes najbolj razširjeni protokol povezavne plasti je protokol Ethernet[1]. Poznamo še protokola ATM in Frame Relay (FR)[2], ki sta bila priljubljena v preteklosti.

Omrežna plast predstavlja obseg celotnega omrežja. Glavna naloga omrežne plasti je usmerjanje paketov proti cilju. Omrežna plast skrbi za pravilno potovanje paketov različnih dolžin po različnih poteh in obsega fragmentacijo ter defragmentacijo paketov. Na tej plasti delujejo usmerjevalniki, ki usmerjajo promet in določajo poti v omrežju. Najbolj znan protokol omrežne plasti je protokol IP. Danes se uporablja protokol IP različice 4 (IPv4), kmalu pa ga bo nadomestila različica 6 (IPv6).

Transportna plast skrbi za prenos sporočila kot celote, vzpostavi povezavo med končnima točkama v omrežju (tj. računalnikoma), nadzoruje tok podatkov in trajanje podatkovne povezave. Protokola transportne plasti sta na primer UDP in TCP. Protokol UDP je nepovezovalni protokol transportne plasti, ki ne zagotavlja zanesljivega prenosa podatkov. V nasprotju s protokolom UDP je protokol TCP povezavni protokol, ki zagotavlja zanesljiv prenos podatkov.

Plast seje vzpostavi, vzdržuje in prekine komunikacijsko sejo med aplikacijama. V primeru izgubljene povezave vzpostavi novo povezavo in prenos nadaljuje v točki, kjer je bil prekinjen.

Predstavitvena plast je del operacijskega sistema in omogoča pretvarjanje podatkov v tako obliko, da bodo uporabni v trenutnem računalniškem okolju. Naloge predstavitvene plasti so stiskanje, raztezanje, šifriranje, dešifriranje itd.

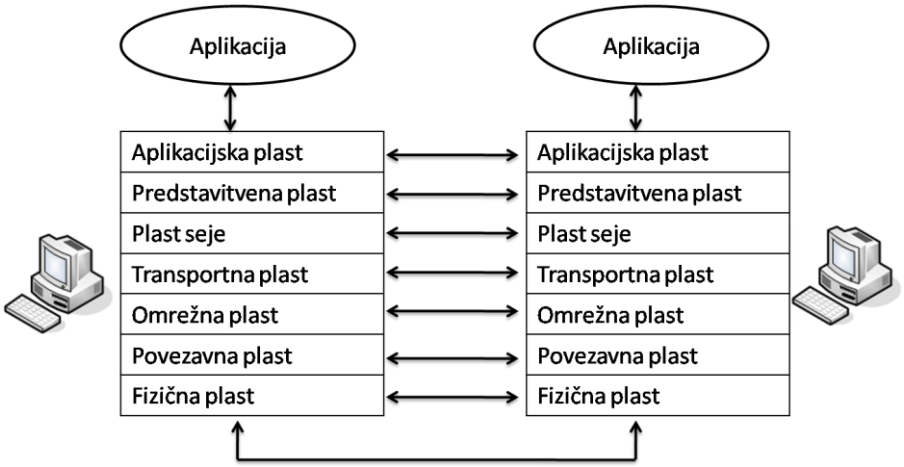
Aplikacijska plast je vmesnik med končnim uporabnikom in komunikacijskim omrežjem. Aplikacijska plast sodeluje z aplikacijami (spletni brskalnik, elektronska pošta itd.), ki predstavljajo podatke in le-te prevzemajo od ostalih uporabnikov. Najbolj znani protokoli aplikacijske plasti so HTTP, FTP in SNMP.

---

[1] Ethernet je definiran po standardu IEE 802.3 in je najpopularnejši protokol za lokalna omrežja.
[2] ATM (ang. Asynchronous Transfer Mode) in Frame Realy sta tehnologiji za napreden način preklapljanja
[2] ATM (ang. Asynchronous Transfer Mode) in Frame Realy sta tehnologiji za napreden način preklapljanja paketov.

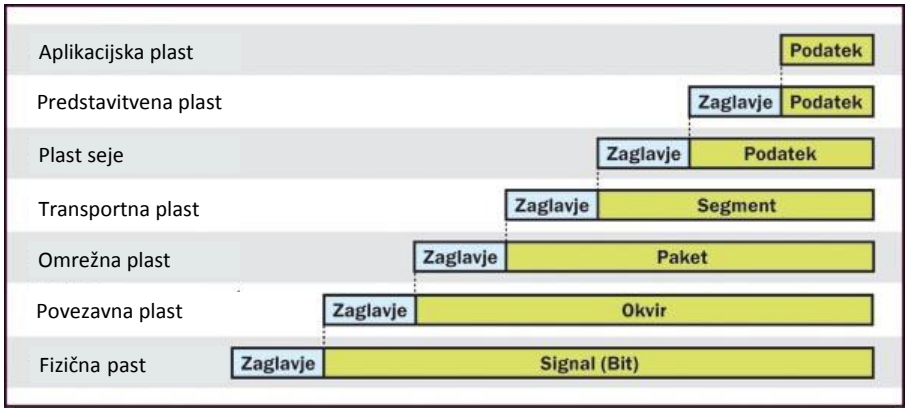
2.1.2 Model OSI in komunikacija med napravami

Če želimo vzpostaviti komunikacijo med dvema napravama (npr. računalnikoma), se moramo držati določenih pravil. Osnovni standard, ki predpisuje ta pravila in po katerem poteka komunikacija med dvema računalnikoma, je referenčni model OSI. Podatek, ki se prenaša iz aplikacije na enem računalniku v aplikacijo na drugem računalniku, potuje skozi sedem plasti modela OSI, kot je prikazano na sliki 2.



Slika 2: Komunikacija med dvema računalnikoma

Vsaka komunikacija med dvema računalnikoma se začne v aplikacijski plasti oddajnika, ko pošljemo podatek v omrežje. Podatek potuje navzdol po protokolnem skladu modela OSI. Pri tem ga vsaka izmed plasti uokviri s krmilnimi podatki oziroma z glavo (ang. header). Vsaka plast podatek obravnava v skladu s svojo funkcijo v modelu, kot na primer podatek na aplikacijski plasti, segment na transportni plasti, paket na omrežni plasti itd. Postopek se imenuje enkapsulacija. Tako je v končni fazi podatek ovit v sedem okvirov. To prikazuje slika 3. Na prejemnikovi strani se postopek ponovi v nasprotnem vrstnem redu, kjer se v procesu deenkapsulacije na vsaki izmed plasti odstrani eden izmed okvirov. Prejemnikova aplikacija sprejme izvirne podatke oddajnika na sedmi (aplikacijski) plasti [10].



Slika 3: Vsaka plast podatkovnemu okviru pripne še svoje zaglavje [11]

2.2 Referenčni model TCP/IP

Referenčni model TCP/IP je postal za razliko od referenčnega modela OSI de facto standard. Nastal je leta 1979 na ameriškem ministrstvu za obrambo za potrebe njihovega omrežja ARPANET. Referenčni model TCP/IP se imenuje po najpomembnejših protokolih TCP in IP. Razdeljen je na štiri plasti, kot je prikazano na sliki 1. Aplikacijska in transportna plast referenčnega modela TCP/IP funkcionalno ustrezata enakima plastema v referenčnem modelu OSI. Internetna plast referenčnega modela TCP/IP ustreza omrežni plasti referenčnega modela OSI. Spodnji dve plasti referenčnega modela OSI sta združeni v eno plast referenčnega modela TCP/IP, ki pomeni povezavo med računalnikom in omrežjem. Plast seje in predstavitvena plast referenčnega modela OSI nista realizirani v modelu TCP/IP.

2.3 Omrežje z internetnim protokolom (IP)

V omrežju se najpogosteje uporablja protokol IP za prenos podatkov znotraj posameznih omrežij in med omrežji, ki temeljijo na različnih prenosnih tehnologijah. Omrežja z internetnim protokolom so postala nekakšen konvergenčni sloj za storitve (aplikacije) nad njim in za omrežja (tehnologije) pod njim [4]. V nadaljevanju se osredotočamo na omrežno plast modela OSI, ki zagotavlja usmerjanje paketov v omrežju. Osnovna podatkovna enota na omrežni plasti je paket (IP-paket).

2.3.1 Protokol IP

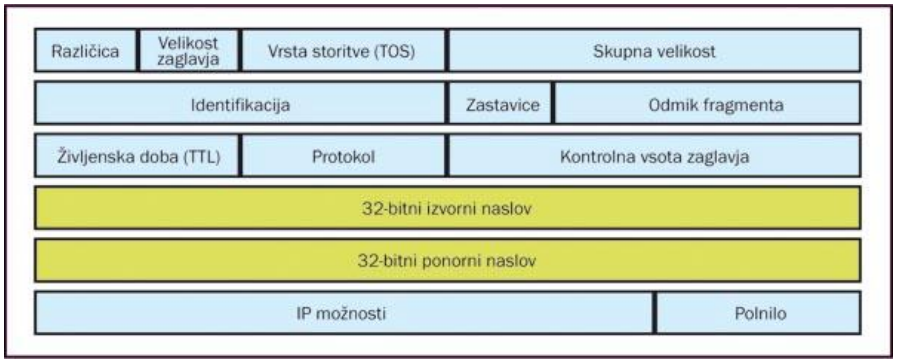
Protokol IP je najpomembnejši komunikacijski protokol. Osnovna naloga protokola IP je usmerjanje prometa od izvora do ponora na podlagi internetnega naslova. Usmerjanje izvaja usmerjevalnik s pomočjo usmerjevalne tabele. Ostale naloge, ki jih zagotavlja protokol IP, so naslavljanje omrežnih naprav, fragmentacija (razgradnja), defragmentacija (ponovno združevanje) in sporočanje o delovanju omrežja IP [22]. Njegova uspešnost temelji predvsem na njegovi robustnosti in preprostosti. Prva različica protokola IP je IPv4 (ang. Internet Protocol Version 4), čeprav se aktivno uvaja protokol IPv6 (ang. Internet Protocol Version 6). Splošno gledano – s protokolom IPv6 se razširi naslovni prostor protokola IPv4 z 32 bitov na 128 bitov.

Protokol IP je nepovezavno orientiran protokol. To pomeni, da se pred prenosom podatkov ne vzpostavi nikakršna povezava med končnima vozliščema. Podatki se preprosto predajo omrežju IP, ki poskrbi za dostavo le-teh na cilj. To pomeni, da se vsak paket v omrežju obravnava neodvisno in ločeno od drugih. S tem pridemo do nezanesljivega delovanja omrežja IP, to pa pomeni, da se paketi na poti lahko izgubijo ali okvarijo [2].

Protokol IP ne vključuje mehanizmov za zanesljiv in kvaliteten prenos po omrežju. Deluje po načelu najboljše možne dostave (ang. best effort) paketov. Omrežje paketov ne bo zavrglo samovoljno, brez pravega vzroka, ampak se bo to zgodilo zaradi preobremenjenosti (zasičenosti) omrežja. V splošnem velja, da so pri manjših obremenitvah omrežja zakasnitve majhne, pri večjih pa postanejo velike in nepredvidljive [22].

2.3.1.1 Paket IP

Skozi omrežje IP se vsaka informacija pošlje kot paket (IP-paket). Paket je zaokrožena celota, ki je sestavljena iz glave paketa (zaglavja) in podatkovnega dela. Na sliki 4 je predstavljena glava paketa.



Slika 4: Glava paketa [11]

Najpomembnejša podatka v glavi paketa sta prejemnikov (izvorni) in pošiljateljev (ponorni) logični naslov. V svojem bistvu IP namreč določa, kam paket potuje in od kod prihaja. Pomembno vlogo ima tudi fragmentacijski bit (ang. fragmentation bit), ki nam pove, ali je paket zaradi združljivosti med heterogenimi omrežji razstavljen na več delov. Protokol IP omogoča fragmentacijo podatkov in tako podpira podatkovno povezavo omrežij z različno maksimalno povezovalno velikostjo (ang. Maximum Transmission Unit – MTU). Pomembno vlogo ima tudi polje ToS (ang. Type of Service), s katerim se paketi razvrščajo glede na zahtevano kakovost storitev [11].

2.3.1.2 Naslavljanje v omrežju IP

Paketi se prenašajo od izvorne do ciljne naprave na podlagi naslovov. Vsaka naprava v omrežju IP ima unikaten naslov. Naslov IP je dolg 32 bitov in je sestavljen iz štirih 8-bitnih polj, imenovanih okteti. Okteti so ločeni s piko in predstavljajo decimalno število med 0 in 255. Naslov IP je razdeljen na omrežni del in del naprave. Mejo med tema dvema deloma določa omrežna maska (ang. netmask). Za primer si poglejmo naslov IP 192.168.12.55, ki ima omrežno masko 255.255.0.0. To pomeni, da prvi del naslova 192.168 označuje omrežje, zadnji del 12.55 pa napravo v tem omrežju.

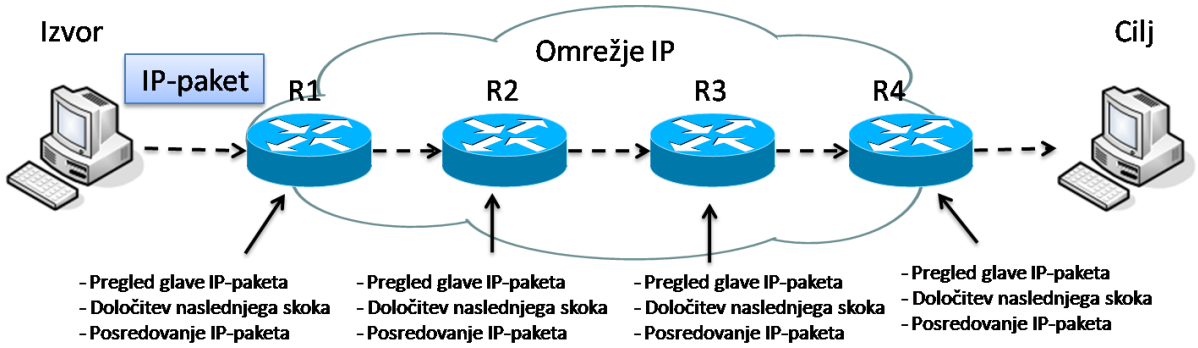
2.3.1.3 Usmerjanje v omrežju IP

Usmerjanje (ang. routing) je proces vodenja paketa od njegovega vira, skozi medomrežja (npr. internet ali drugo prostrano omrežje) do njegovega končnega cilja [24]. Usmerjanje se izvaja za vsak paket posebej. Usmerjanje je sestavljeno iz dveh osnovnih aktivnosti: določanja poti usmerjanja in posredovanja paketov skozi omrežje. Informacijo o usmerjanju vsebujejo usmerjevalne tabele. Te informacije so odvisne od usmerjevalnega algoritma, ki ga uporablja usmerjanje. Cilj usmerjevalnega algoritma je izbira najboljše poti. Usmerjanje lahko poteka statično in dinamično [14].

Statično usmerjanje se izvaja na osnovi v naprej nastavljene usmerjevalne tabele, v katero so vneseni podatki o poti. Izvajanje po nastavljeni poti se nadaljuje, dokler ne vnesemo podatka o spremembi poti. S statičnim usmerjanjem se omrežje ne more odzivati na napake v omrežju. V primeru napake na usmerjevalniku postane pot do končnega cilja nedefinirana. Napaka je lahko posledica izgube podatka v usmerjevalni tabeli za določene segment oziroma napaka v komunikacijski povezavi.

Dinamično usmerjanje uporablja usmerjevalne protokole za avtomatično osveževanje usmerjevalnih tabel. Pri dinamičnem usmerjanju se usmerjevalne tabele ustvarijo samodejno. Z dinamičnim usmerjanjem najdemo najboljše poti v omrežju. Izberemo jih glede na usmerjevalne metrike, kot so na primer pasovna širina, cena povezave, zakasnitev in število skokov.

Usmerjanje deluje po načelu "skok-za-skokom" (ang. hop-by-hop). To pomeni, da se pot za prenos paketov skozi omrežje določa sproti. Usmerjevalnik z uporabo usmerjevalne tabele določi naslednji skok (usmerjevalnik) za vsak IP-paket posebej. IP-paket se usmerja na podlagi ciljnega naslova v glavi IP-paketa. Vsakemu IP-paketu se na vsakem usmerjevalniku (R1, R2, R3, R4) pregleda glava IP-paketa. Na podlagi usmerjevalne tabele se določi naslednja naprava. Proces pregledovanja glave in določanja naslednje naprave, kamor se posreduje IP-paket, se ponavlja, dokler IP-paket ne prispe do cilja. Postopek preprostega usmerjanja in pot (črtkana črta), po kateri potuje IP-paket, prikazuje slika 5.



Slika 5: Preprosto usmerjanje v omrežju IP

2.3.1.4 Usmerjevalnik

Usmerjevalnik je naprava, ki povezuje dve ali več omrežij ter tvori medomrežje. Naloga usmerjevalnika je zbiranje in vzdrževanje informacij o topologiji omrežja. Na podlagi teh informacij usmerja izbrane pakete na izbrane naslove. Svoje naloge opravlja na omrežni plasti v referenčnem modelu OSI. Usmerjevalnik pri posredovanju prometa proti cilju uporablja ciljni naslov in usmerjevalne tabele. Tabele se lahko nastavijo ročno, lahko pa se zgradijo samodejno s pomočjo dinamičnih usmerjevalnih protokolov. Vsak usmerjevalnik v omrežju IP opravlja proces usmerjanja za vsak paket posebej.

2.3.1.5 Usmerjevalna tabela

Usmerjevalna tabela je spisek "najboljših poti" (pogosto edinih poti). V tabeli je shranjena informacija o omrežjih, ki so neposredno priključena na vmesnike usmerjevalnika. Tabela vsebuje informacijo o poteh do omrežij, ki jih je administrator bodisi vpisal ročno (statično) bodisi se jih je usmerjevalnik "naučil" s pomočjo dinamičnih usmerjevalnih protokolov [24]. Usmerjevalna tabela je shranjena v pomnilniku usmerjevalnika. Usmerjevalnik shrani v usmerjevalno tabelo najboljše razpoložljive povezave, ki jih sporoča usmerjevalni protokol. Na podlagi informacij v usmerjevalni tabeli usmerjevalnik usmerja pakete. Usmerjevalne tabele so lahko zgrajene s pomočjo usmerjevalnih protokolov.

2.3.2 Usmerjevalni protokoli

Usmerjevalni protokol gradi in dopolnjuje (osvežuje) usmerjevalne tabele. Usmerjevalni protokoli ne sodelujejo pri prenosu podatkov. Njihova naloga je ažuriranje in zapisovanje informacij o povezavah v omrežju. Informacije o povezavah se razširjajo med usmerjevalniki in so potrebne za izračun optimalne poti v omrežju. Usmerjevalni protokoli izvajajo usmerjevalne algoritme, ki izračunajo ustrezne poti v omrežju.

Usmerjevalne protokole delimo na notranje usmerjevalne protokole (ang. Interior Gateway Protocol – IGP) in zunanje usmerjevalne protokole (ang. Exterior Gateway Protocol – EGP). V notranjih omrežjih, kot so lokalna omrežja, se izvajajo usmerjevalni protokoli IGP, v zunanjih omrežjih pa usmerjevalni protokoli EGP. Najbolj znana notranja usmerjevalna protokola, ki temeljita na osnovi stanja povezav (ang. link state), sta OSPF (ang. Open Shortest Path First). Primer zunanjega usmerjevalnega protokola je BGP (ang. Border Gateway Protocol).

2.3.2.1 Protokol OSPF

Protokol OSFP je najbolj razširjen notranji usmerjevalni protokol v omrežjih IP. Razvit je bil s strani IETF (ang. Internet Engineering Task Force) zaradi pomanjkljivosti protokola RIP (ang. Routing Information Protocol). Glavni pomanjkljivosti protokola RIP sta premajhna robustnost in neučinkovitost pri iskanju alternativnih poti po omrežju. Protokol OSPF deluje znotraj avtonomnega sistema – usmerjevalne domene. Zbira podatke o stanju povezav med usmerjevalniki in shranjuje topologijo omrežja. Za delovanje uporablja algoritem SPF (ang. Shortest Path First). OSPF se na spremembe v omrežju odziva hitro in učinkovito, ker takoj sporoči spremembo sosednjim usmerjevalnikom in pošlje le del (ne celotne) usmerjevalne tabele, kjer so se zgodile spremembe. Omrežje OSPF se lahko razdeli na usmerjevalna področja (ang. area). S tem se poenostavi administracija, optimizira promet ter zmanjša obremenjenost resursov. Področja se identificirajo z 32-bitnimi števili, kot je to v primeru naslova IPv4. Protokol OSPF za izmenjavo podatkov o stanju povezav ne uporablja transportnega protokola TCP kot na primer protokol BGP. OSPF uporablja naslov za oddajanje več prejemnikom (ang. multicast) zaradi prenosa poti na povezavah razpršenega oddajanja. Pri protokolu OSPF je za komunikacijo med usmerjevalniki mogoče vklopiti dodatno avtentikacijo. Tako se lahko samo znani usmerjevalniki z uporabo gesla vključijo v usmerjanje OSPF [27].

2.3.2.2 Protokol BGP

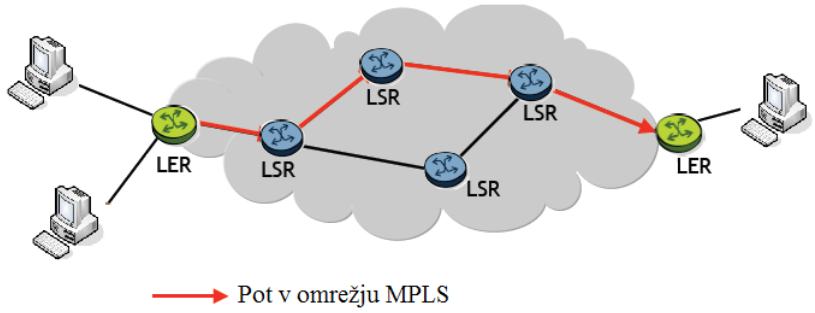
Protokol BGP je zunanji usmerjevalni protokol, ki se uporablja za komunikacijo med avtonomnimi sistemi. Avtonomni sistem je skupek naprav (npr. usmerjevalnikov). Protokol BGP je opisan kot protokol, ki uporablja vektor poti, s katerim pozna celotno omrežje. Na osnovi vektorja poti določi najustreznejšo povezavo med omrežji. Protokol BGP se o najboljših poteh odloča na osnovi parametra pot, ki je zapis avtonomnih sistemov za določeno potovanje smeri. Manjše kot je število zapisov v parametru poti, krajša je pot do cilja. Sistemi BGP si med seboj izmenjujejo omrežne informacije, ki vključujejo popolne poti avtonomnih sistemov. Promet mora prepotovati popolne poti avtonomnih sistemov, da doseže definirane cilje (določena omrežja) [26].

2.3.3 Problem omrežij z internetnim protokolom

Pri preprostih omrežjih z internetnim protokolom proces usmerjanja ne predstavlja nobenega problema. Problem se pojavi pri velikih količinah omrežnega prometa in kompleksnih omrežjih, ker se mora pregledovati glava IP-paketa na vsakem vozlišču. Pot za prenos IP-paketov namreč ni določena vnaprej in se vzpostavlja sproti. Za vsak IP-paket, ki potuje skozi omrežje, mora usmerjevalnik izvesti enak proces za iskanje naslednjega vozlišča. To je izredno kompleksna in potratna procedura, ki za posredovanje IP-paketov porabi več časa in tako zmanjšuje zmogljivost omrežij. Omrežje s protokolom IP je nezanesljivo, nepovezavno orientirano paketno omrežje, ki ga je mogoče optimizirati tako, da v omrežje IP vpeljemo tehnologijo MPLS.

3 MPLS (ang. Multiprotocol Label Switching)

MPLS je tehnologija (protokol), ki v obstoječa nepovezavno usmerjena omrežja IP vpelje povezavno usmerjen princip. Povezavno usmerjen princip pomeni, da se pred prenosom podatkov vzpostavi povezava (pot) med končnima vozliščema. Na sliki 6 je prikazana vzpostavljena pot skozi omrežje MPLS z rdečo barvo. Povezava se vzpostavi za vsak prenos podatkov skozi omrežje.



Slika 6: Vzpostavitev povezave v omrežju MPLS [25]

Za prenos podatkov po vnaprej vzpostavljeni poti je potrebna labela (labela MPLS). Labela se dodeli vsakemu IP-paketu v vhodnem vozlišču na robu omrežja MPLS. Znotraj omrežja se potem IP-paketi posredujejo le na osnovi labele, ki se zamenja v vsakem vozlišču. Tako ni potreben vpogled v glavo IP-paketov v vsakem vozlišču na njegovi poti. Z labelo se prihrani čas, ki bi ga potrebovalo vozlišče za analizo glave IP-paketa. Prenos podatkov postane hitrejši, saj s tem odpadeta "dolgotrajno" procesiranje in usmerjanje IP-paketa v vsakem vozlišču.

Tehnologija MPLS ne nadomešča klasičnega usmerjanja IP, ampak deluje z obstoječimi usmerjevalnimi tehnologijami (protokoli). To pomeni, da v obstoječo infrastrukturo omrežja IP lahko vpeljemo tehnologijo MPLS, ki bo omogočala hitro posredovanje IP-paketov na osnovi label. Omrežja, kjer je izveden MPLS, se imenujejo omrežja IP/MPLS oziroma omrežja MPLS. MPLS je tehnologija, ki združuje zmogljivost in enostavnost usmerjanja omrežne plasti skupaj z visoko hitrostjo preklapljanja povezavne plasti. Z MPLS pohitrimo omrežni tok prometa IP-paketov in omogočimo funkcionalnosti, ki so bile v omrežju IP težje izvedljive. To so predvsem prometni inženiring, navidezna zasebna omrežja in kakovost storitev, ki bodo opisani v nadaljevanju.

3.1 Uvrstitev MPLS v referenčni model OSI

Protokol MPLS je neodvisen od protokolov omrežne in povezavne plasti, zato ga ne uvrščamo v nobeno izmed plasti referenčnega modela OSI. Protokol MPLS uvrstimo med omrežno in povezavno plastjo modela OSI ter ga obravnavamo kot protokol "2.5-plasti", kar nazorno prikazuje slika 7.

7	Aplikacijska plast
6	Predstavitvena plast
5	Plast seje
4	Transportna plast
3	Omrežna plast
2.5	MPLS
2	Povezavna plast
1	Fizična plast

Slika 7: MPLS v referenčnem modelu OSI

MPLS ni protokol omrežne plasti, ker nima lastnega usmerjanja in naslavljanja. MPLS uporablja IP-naslavljanje in usmerjanje (z razširitvami). Glede na standarde so mogoči tudi drugi tipi naslavljanj in usmerjanj, vendar je praktično v uporabi samo IP. Prav tako MPLS ni protokol povezavne plasti, ker lahko deluje prek različnih povezavnih tehnologij, kot sta na primer Ethernet in ATM. MPLS ne predstavlja samostojne plasti v referenčnem modelu OSI, ker nima enega samega formata [5].

3.2 Arhitektura MPLS

3.2.1 Ločitev kontrolne in podatkovne ravnine

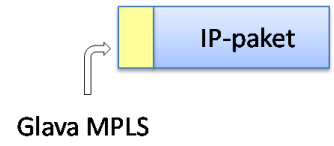
V usmerjevalniku IP sta funkciji posredovanja in kontrole združeni, zato je glavna ideja protokola MPLS ločitev teh dveh funkcij. Protokol MPLS razdeli podatkovno (posredovalno) in kontrolno (signalizacijsko) ravnino, to pa omogoča, da se oba segmenta razvijata neodvisno drug od drugega.

Kontrolna ravnina skrbi za izmenjavo label in usmerjevalnih informacij med sosednjimi vozlišči. Sestavljena je iz signalizacijskih in usmerjevalnih protokolov. Signalizacijski protokoli kontrolne ravnine skrbijo za izmenjavo label in vzpostavljanje ter vzdrževanje poti v omrežju MPLS. V tej ravnini se iščejo najboljše poti po omrežju. Kontrolna ravnina si z uporabo standardnih usmerjevalnih protokolov izmenjuje informacije med usmerjevalniki, ki jih uporabi za gradnjo in vzdrževanje usmerjevalnih tabel.

Podatkovna ravnina je preprost mehanizem za posredovanje dejanskega prometa (IP-paketov). Za IP-pakete, opremljene z labelo, se v podatkovni ravnini uporablja tabela LIB (ang. Label Information Base), s pomočjo katere se izvaja proces zamenjave label. Paket, ki ni opremljen z labelo, se obdela kot normalen IP-paket s pomočjo tabele FIB (ang. Forwarding Information Base).

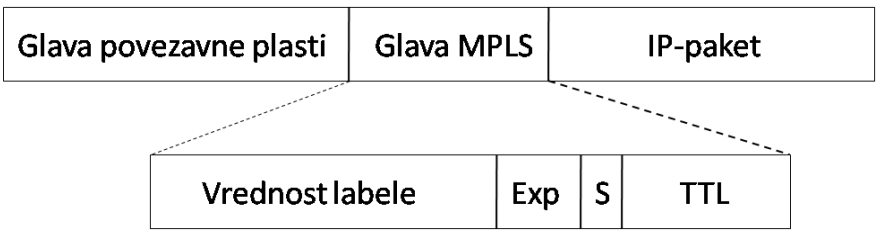
3.2.2 Glava MPLS

Glava MPLS se doda vsakemu IP-paketu (slika 8), ki vstopi v omrežje MPLS. Proces dodajanja glave MPLS se imenuje ovijanje (enkapsulacija) IP-paketa po protokolu MPLS.



Slika 8: Dodajanje glave MPLS

Glava MPLS se nahaja med glavo povezavne plasti in IP-paketom. V glavi MPLS je labela. Vsebino glave MPLS in polja labele prikazuje slika 9. Glava MPLS se IP-paketu odstrani na izhodu iz omrežja MPLS.



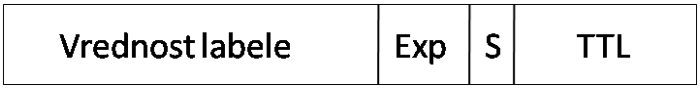
Slika 9: Glava MPLS

3.2.3 Labela MPLS

Labela je kratek identifikator z 32-bitno fiksno dolžino in identificira pot, po kateri bo paket potoval. Labela, ki je dodana na začetek paketa, omogoča hitro posredovanje paketov. Na osnovi label strojna oprema hitro preklaplja pakete med povezavami. Labela je veljavna samo lokalno, to pomeni, da je uporabna in primerna za posamezno povezavo med sosednjimi vozlišči. Nekatere storitve, ki temeljijo na protokolu MPLS, potrebujejo več kot eno labelo za posredovanje paketa, opremljenega z labelo. Za te storitve se uporablja sklad label.

Labela je zgrajena iz štirih polj (slika 10):

- **vrednosti labele**, ki je shranjena v polju z dolžino 20 bitov;
- **polja Exp** z dolžino treh bitov, ki so namenjeni za dodeljevanje razreda storitev (ang. Class of Service – CoS). S tem poljem se zagotovi kakovost storitev;
- **bita S** (ang. Stacking), ki označuje hierarhičnost sklada label. Bit S se nastavi na 1, če je labela na dnu sklada label (zadnja labela v skladu), sicer se nastavi na 0;
- **polja TTL** (ang. Time To Live) z dolžino osmih bitov, ki predstavlja življenjsko dobo paketa. Polje TTL ima enako vlogo kot polje TTL v glavi IP-paketa, poleg tega pa je namenjeno prepričevanju zankanja v omrežju MPLS. Pri prehodu paketa skozi usmerjevalnik se vrednost TTL zmanjša za 1. Usmerjevalnik zavrže paket, ko vrednost TTL pade na 0.

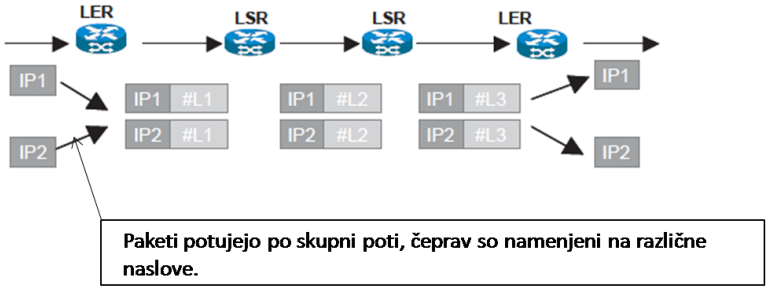


Slika 10: Zgradba labele MPLS [1]

Tehnologija povezavne plasti kot je Ethernet ne podpira label, zato je labela vstavljena v glavo MPLS, ki se nahaja med glavo protokola povezavne plasti in glavo protokola IP. Če tehnologija povezavnega ali fizičnega sloja podpira labele kot na primer tehnologija ATM, se labela MPLS zapiše (enkapsulira) na mesto originalne labele.

3.2.4 Ekvivalentni razred posredovanja

Ekvivalentni razred posredovanja (ang. Forwarding Equivalence Class – FEC) je skupina paketov, ki imajo enake pogoje za prenos skozi omrežje. To pomeni, da se paketi posredujejo po isti poti, obravnavajo se na enak način in so opremljeni z enako labelo. Primer delovanja je prikazan na sliki 11. Za razliko od usmerjanja IP se pri MPLS dodelitev paketa k določeni FEC skupini zgodi le enkrat, to je ob vstopu paketa v omrežje. Skupina FEC temelji na potrebnih pogojih prenosa za določeno skupino paketov ali za določen naslov [15].

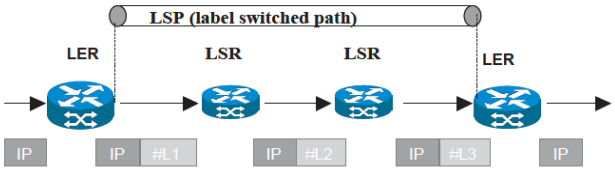


Slika 11: Primer ekvivalentnega razreda posredovanja [1]

Prednost dodeljevanja posamezne labele različnemu toku paketov z enakim FEC-om je združevanje prometa, s katerim zmanjšamo število label in količino kontrole distribucije label. Tako izboljšamo skalabilnost in zmanjšamo potrebne resurse CPE (Centralne procesne enote).

3.2.5 Pot LSP

Pot LSP (ang. Label Switched Path – LSP) je pot skozi omrežje MPLS. Vzpostavi se med začetnim in končnim usmerjevalnikom ter služi za prenos IP-paketov na osnovi label v omrežju MPLS. Za prenos IP-paketov po vnaprej vzpostavljeni poti se uporablja labela, ki se zamenja pri vsakem prehodu čez usmerjevalnik. Tako je pot LSP pravzaprav zaporedje usmerjevalnikov, ki so na poti IP-paketa. Pot LSP je enosmerna povezava (tunel) med vhodnim in izhodnim usmerjevalnikom (slika12). Za dvosmerno povezavo je treba vzpostaviti dve poti.



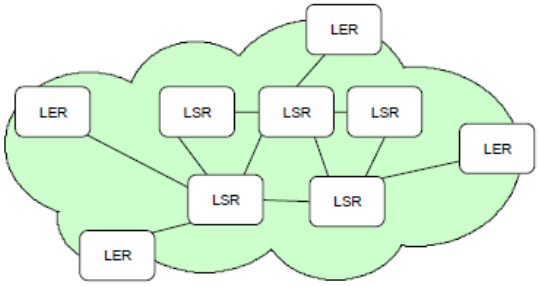
Slika 12: Pot LSP [1]

Ob oddaji podatkov ali detekciji toka podatkov se vzpostavi pot LSP s signalizacijskim protokolom in na osnovi kriterija FEC. Poznamo dva načina vzpostavljanja poti. Prvi način je s signalizacijskim protokolom LDP, pri katerem se pot določi po skokih, na osnovi usmerjevalne tabele. Drugi način je vzpostavljanje poti z upoštevanjem omejitev, kot so pasovna širina, zahteva po kakovosti storitev in administrativna politika. Te poti se imenujejo eksplicitne poti in se vzpostavijo s signalizacijskima protokoloma, kot sta RSVP in CR-LDP.

3.2.6 Usmerjevalniki v omrežju MPLS

Protokol MPLS je primeren predvsem za hrbtenična omrežja IP. Naprave, s katerimi zgradimo omrežje MPLS, se imenujejo usmerjevalniki LSR (ang. Label Switch Router). Vsak usmerjevalnik LSR uporablja za pošiljanje klasičnih IP-paketov tabelo FIB, za pošiljanje IP-paketov, opremljenih z labelo, pa tabelo LIB. Usmerjevalnik LSR predstavlja funkcijo usmerjevalnika in stikala obenem ter omogoča posredovanje IP-paketov skozi omrežje MPLS. Usmerjevalniki LSR sodelujejo pri izmenjavi label ter vzpostavljanju in izgradnji poti LSP. Glede na različne naloge, ki jih opravljajo, se določi njihov položaj v omrežju MPLS.

Usmerjevalnike LSR na splošno delimo na vhodne, vmesne in izhodne. Vhodni in izhodni LSR sta robna usmerjevalnika LER (ang. Label Edge Router), ker sta na samem robu omrežja. Jedro omrežja sestavljajo vmesni usmerjevalniki LSR. Usmerjevalniki LSR izvajajo operacije dodajanja (ang. push), zamenjave (ang. swap) in odstranitve (ang. pop) label. Osnovna razporeditev usmerjevalnikov je prikazana na sliki 13.



Slika 13: Osnovna razporeditev usmerjevalnikov [6]

Vhodni usmerjevalnik sprejme IP-paket. IP-paketu doda labelo MPLS in ga posreduje do vmesnega usmerjevalnika.

Vmesni usmerjevalnik zmanjša vrednost TTL za 1 in zamenja labelo IP-paketa, ter ga posreduje do naslednjega usmerjevalnika vzdolž poti LSP. Naloga vmesnega usmerjevalnika LSR je zamenjava vhodne labele z izhodno in pošiljanje IP-paketa glede na labelo.

Izhodni usmerjevalnik odstrani IP-paketu labelo MPLS in ga usmeri naprej na podlagi ciljnega naslova IP. Zaradi posredovanja IP-paketa na osnovi label se močno zmanjšajo procesorske zahtevnosti usmerjevalnikov LSR.

3.3 Signalizacijski protokoli

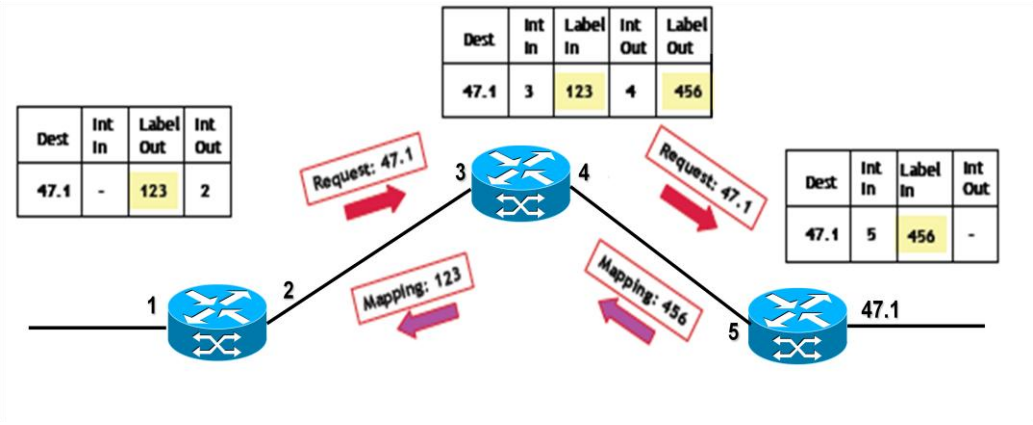
Signalizacijski protokoli so protokoli signalizacijske ravnine. Vzpostavljajo poti LSP in določajo vrednosti label na dotičnih poteh. Poznamo različne signalizacijske protokole, kot so LDP (ang. Label Distribution Protocol), RSVP (ang. Resource Reservation Protocol) in CR-LDP (ang. Constraint based Label Distribution Protocol). LDP deluje po skokih iz vozlišča na naslednje vozlišče in izbere iste fizične poti kot protokol IGP (npr. protokol OSPF). Protokol CR-LDP je razširitev protokola LDP s podporo eksplicitno določenim potem. Protokol RSVP se veliko uporablja v operaterskih okoljih, saj omogoča razširitev podpore prometnega inženiringa in eksplicitno določene poti.

3.3.1 Protokol LDP

Protokol LDP se izvaja v kontrolni ravnini in je namenjen za distribucijo label in vzpostavljanje poti LSP. Protokol LDP vzpostavi pot za oddajo podatkov v eno smer. Vzpostavljena pot je optimalna pot, ki jo izbere protokol LDP. Slednja pa je enaka poti, ki bi jo izbral tudi protokol IGP. Motivacija za vzpostavitev poti LSP s protokolom LDP je torej v hitrem posredovanju IP-paketov med usmerjevalniki. Za vsako vzpostavljeno pot se dodeli labela, s pomočjo katere se posredujejo IP-paketi na ciljni naslov IP. Vsaka ustvarjena pot LSP se vključi v razred FEC, kateremu se pripiše določena labela.

Protokol LDP uporablja za dodeljevanje label dve signalizacijski sporočili:

- **Zahteva po labeli (ang. Label Request)** – s tem sporočilom usmerjevalnik LSR zahteva labelo od naslednjega usmerjevalnika v smeri toka podatkov.
- **Preslikava labele (ang. Label Mapping)** – je odgovor na zahtevo po labeli in vsebuje vrednost labele, ki se shrani v tabelo LIB.



Slika 14: Razširjanje signalizacijski sporočil [25]

S signalizacijskimi sporočili dodelimo vsakemu usmerjevalniku labelo, ki se shrani v tabelo LIB, kot je prikazano na sliki 14. Z dodelitvijo label na vsakem usmerjevalniku je vzpostavljena pot, po kateri se bodo posredovali IP-paketi. Pri posredovanju IP-paketov se uporabljajo labele, ki so shranjene v tabeli LIB.

S protokolom LDP se izmenjujejo informacije o vezanih labelah med dvema usmerjevalnikoma. Za izmenjavo informacij med dvema soležnima usmerjevalnikoma se vzpostavi LDP-seja. Protokolu LDP se lahko dodajo določene omejitve pri vzpostavljanju poti, pri čemer dobimo razširjeni protokol CR-LDP. Protokol CR-LDP vzpostavi pot, ko se serija sporočil po zahtevah label prenese od vhodnega do izhodnega usmerjevalnika LSR. Če zahtevana pot ustreza omejitvam (npr. zadostno število primernih omrežnih virov), se labele razširijo in načrtno delijo s pomočjo sporočil za razširjanje label.

3.3.2 Protokol CR-LDP

Protokol CR-LDP je množica postopkov, ki omogočajo vozlišču poleg izmenjave label in vzpostavitve poti LSP tudi usmerjanje z omejitvami. Omejitve, ki se upoštevajo pri usmerjanju, so topologija omrežja, zakasnitev in pasovna širina povezav itd. Na osnovi omejitev se izbere pot, po kateri se posredujejo paketi. Protokol CR-LDP vsebuje del funkcionalnosti protokola LDP in dodatek, ki omogoča vzpostavljanje eksplicitnih poti, prenos prometnih parametrov (npr. zahteve po pasovni širini) za rezervacijo virov in opcije za zaščito poti. CR-LDP je protokol trdega stanja (ang. hard state), saj se prenos signalizacijskih sporočil prenese enkrat brez kasnejšega osveževanja [4].

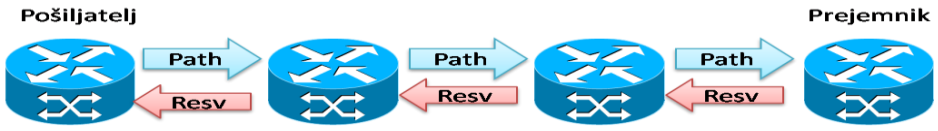
Transportni mehanizem za iskanje sosedov, tako imenovanih "peerov", je protokol UDP, za vsa ostala sporočila pa se uporablja protokol TCP. Vzpostavitev eksplicitne poti se začne s sporočilom po zahtevi labele, ki vsebuje spisek vozlišč, med katerimi želimo vzpostaviti eksplicitno pot. Signalizacijsko sporočilo za zahtevo labele potuje po izbrani poti do končnega oziroma ciljnega vozlišča. Vzdolž poti se s sporočilom preverja razpoložljivost omrežnih virov. Če so omrežni viri na voljo, se labele dodelijo in razširjajo od končnega do začetnega

vozlišča s sporočilom o preslikavi labele. Tako se lahko samo v enem obhodu signalizacijskih sporočil vzpostavi eksplicitna pot. S protokolom CR-LDP lahko vzpostavljamo tako striktno kot ohlapno določene poti ter alternativne poti z možnostjo ponovnega usmerjanja in optimizacije poti [4].

3.3.3 Protokol RSVP

Protokol RSVP je signalizacijski protokol, ki se uporablja pri oddajanju prometnih tokov enemu (unicast) ali več uporabnikom (multicast). Načrtovan je bil za rezervacijo omrežnih virov, ki so pomemben del prometnega inženiringa. Rezervacije omogočajo podporo eksplicitno določenim potem.

S protokolom RSVP se vzpostavi pot LSP za enosmerne prometne tokove, ki zahtevajo določene omrežne vire. Za rezervacijo omrežnih virov se uporabljata sporočili "zahteva" (Path) in "odgovor" (Resv), kot je prikazano na sliki 15. Sporočilo "Path" potuje od začetnega usmerjevalnika (pošiljatelja), preko vmesnih, do končnega usmerjevalnika (prejemnika). Z uspešnim prihodom do končnega usmerjevalnika se sproži sporočilo "Resv", ki se prenese do vhodnega usmerjevalnika. Tako se opravi rezervacija in vzpostavi se pot LSP za prometni tok. Če manjka omrežnih virov, lahko usmerjevalnik zavrne zahtevo po vzpostavitvi poti.



Slika 15: Potek rezervacije resursov

Protokol RSVP je protokol mehkega stanja (ang. soft-state). To pomeni, da se za vzdrževanje vzpostavljenih poti zahteva osveževanje. Ta lastnost omogoča samodejno prilagajanje na spremembe v omrežju.

3.4 Distribucija label

Labele so lokalne za vsak par usmerjevalnikov in nimajo globalnega pomena skozi omrežje MPLS. Za dodeljevanje in razširjanje label se lahko uporabita dva načina [4]:

- kontrolno voden način (ang. control-driven),
- podatkovno voden način (ang. data-driven).

V podatkovno vodenem načinu se labele dodelijo in razširjajo šele, ko paket prispe do vozlišča. Pot LSP se tako vzpostavi s prihodom prvega paketa v vozlišče oziroma ob prihodu določenega števila paketov. Prednost tega načina je, da se labele določajo in razširjajo pri dejanskem prometu paketov, saj ne obremenjujejo povezav, če ni potrebe po pošiljanju paketov. To je dobro predvsem za kratke pretoke paketov. Pri velikem številu kratkih pretokov paketov se močno obremeni omrežje, zato se pojavi problem razširljivosti podatkovno vodenega načina. Zaradi tega se je uveljavil kontrolno vodeni način. Dodeljevanje in razširjanje label po tem načinu poteka na podlagi usmerjevalnih informacij. Tako se dodeljevanje in razširjanje label izvedeta pred prihodom paketov v vozlišče [5].

Ustrezni mehanizem, ki nam pove, katero labelo naj uporabijo sosednji usmerjevalniki, je protokol za distribucijo label LDP. Protokol LDP uporabljajo usmerjevalniki omrežja MPLS za medsebojno obveščanje o preslikavah med labelami in pretoki.

Dva sosednja usmerjevalnika LSR uporabljata LDP za izmenjavo naslednjih sporočil [15]:

- **Sporočila o odkritju**: odkrivanje prisotnosti sosedov LDP.
- **Sejna sporočila**: vzpostavljanje, vzdrževanje in rušenje LDP-seje.
- **Oglasna sporočila**: ustvari, zahtevaj, preslikaj ali izbriši labele, oglaševanje ali brisanje LDP-vmesnikov.
- **Sporočila za obvestila**: dajanje informacije o nadzoru in napakah.

Za prenos informacij o labelah sosednjim usmerjevalnikom skrbita dva načina za distribucijo label [32]:

- **Promet proti uporabniku na zahtevo**: usmerjevalnik v tem načinu dodeli labelo potem, ko dobi prvi IP-paket, ki ga mora poslati po poti LSP.
- **Promet proti uporabniku brez zahteve**: usmerjevalnik dodeli labelo pred prvim IP-paketom, ki ga mora poslati po poti LSP.

Kontrola nad labelami

S kontrolo nad labelami določimo, kateri usmerjevalnik odloča o dodeljevanju in o razširjanju label. Obstajata dva načina [16]:

- **Neodvisna kontrola:** Pri neodvisni kontroli si vsak usmerjevalnik posebej dodeli labelo za določen FEC, neodvisno od ostalih usmerjevalnikov. Pot LSP se vzpostavi dinamično, tako da se vezava label razširja skozi omrežje ne glede na to, ali je pot LSP vzpostavljena ali ne.
- **Urejena kontrola:** V urejeni kontroli zunanji izhodni usmerjevalnik, ki je zadnji na poti LSP, dodeli labele vsem ostalim usmerjevalnikom. Pot LSP se vzpostavi statično, tako da se vezava label razširja skozi omrežje, preden je vzpostavljena pot LSP.

Shranjevanje label

Za shranjevanje label v usmerjevalniku sta na voljo dva načina [16]:

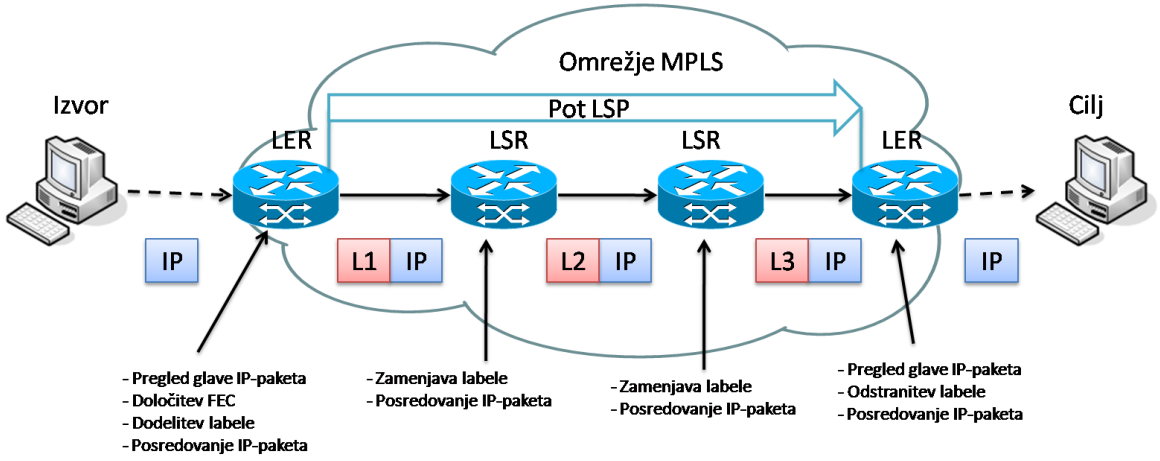
- **Liberalno shranjevanje:** Usmerjevalnik ohrani vse labele, tudi če ni nobenega paketa, ki bi ga bilo treba posredovati skozi omrežje MPLS. V današnjih usmerjevalnikih je ta način najbolj razširjen, saj so usmerjevalniki dovolj zmogljivi in ni potrebe po omejevanju prostora za shranjevanje label.
- **Konzervativno shranjevanje label:** Usmerjevalniki ohranijo le labele, ki se uporabljajo pri posredovanju paketov skozi omrežje MPLS. Labele, ki se ne uporabljajo, se zavržejo. To je pomembno predvsem v napravah, kot so stikala ATM, kjer je prostor za shranjevanje label izredno omejen.

3.5 Delovanje omrežja MPLS

Delovanje omrežja IP s protokolom MPLS temelji na hitrem posredovanju IP-paketov s pomočjo label. IP-paketi se v omrežju MPLS ne usmerjajo več na podlagi ciljnega naslova v glavi IP-paketa, temveč se preklapljajo na osnovi label. Potovanje IP-paketa, opremljenega z labelo, skozi omrežje MPLS se pohitri, ker se analiza glave IP-paketa in dodelitev FEC-a opravljata samo na vhodu omrežja MPLS.

Vhodni usmerjevalnik LER sprejme nelabeliran IP-paket in analizira glavo IP-paketa. Usmerjevalnik LER na podlagi algoritma najboljšega ujemanja pogleda v posredovalno tabelo in določi, kateremu FEC pripada IP-paket. S pomočjo te informacije usmerjevalnik IP-paketu dodeli labelo (L1) in ga posreduje do naslednjega vmesnega usmerjevalnika LSR po vnaprej vzpostavljeni poti LSP. Pot LSP je določena na osnovi usmerjevalne topologije na omrežni plasti. V jedru omrežja usmerjevalniki LSR ne analizirajo glave IP-paketa, temveč posredujejo IP-pakete samo na podlagi labele.

Vmesni LSR prejme IP-paket, ki je opremljen z labelo L1. Posredovalna komponenta s pomočjo informacije iz labele in podatka o vhodnem usmerjevalniku določi izhodni vmesni usmerjevalnik in izhodno labelo L2 na podlagi natančnega ujemanja v posredovalni tabeli. Usmerjevalnik LSR zamenja vhodno labelo z izhodno ter pošlje IP-paket do ustreznega vmesnega usmerjevalnika. To se ponavlja toliko časa, dokler IP-paket, opremljen z labelo, ne prispe do izhodnega usmerjevalnika LER. Posredovalna komponenta izhodnega usmerjevalnika LER odstrani labelo in IP-paket posreduje na podlagi usmerjanja IP. Celotni postopek delovanja prikazuje slika 16.



Slika 16: Delovanje v omrežju MPLS

4 Aplikacije na osnovi protokola MPLS

4.1 Prometni inženiring

Prometni inženiring je mehanizem za upravljanje pretoka podatkov preko omrežja IP. S prometnim inženiringom dosežemo optimalno izrabo omrežnih virov (pasovne širine in povezav) s porazdelitvijo prometa na ostale poti celotnega omrežja IP. Tako optimiziramo delovanje omrežja, zagotovimo zanesljiv in hiter prenos podatkov skozi omrežje ter preprečimo preobremenitev v omrežju. S prometnim inženiringom se izognemo situaciji, ko so nekateri segmenti omrežja zasičeni, drugi segmenti pa neizkoriščeni. Cilj prometnega inženiringa je najbolj učinkovita izraba omrežnih virov, predvsem pasovne širine.

Za učinkovito izvajanje prometnega inženiringa je potrebno upoštevati naslednje komponente:
- razširjanje topoloških informacij,
- izbira in izračun poti,
- usmerjanje prometa prek poti,
- upravljanje prometa.

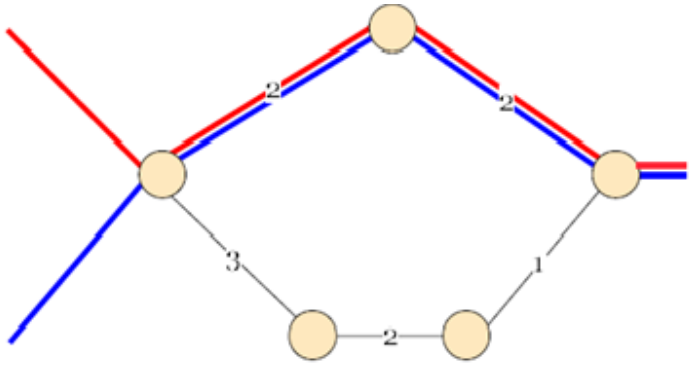
Vsako vozlišče v omrežju si izdela svoj topološki zemljevid omrežja. Ker je omrežje porazdeljen sistem, je pri tem potrebno čim hitrejše razširjanje topolških informacij med vsemi vozlišči. Razširjanje informacij o topologiji omrežja je pomembno predvsem pri izbiri in izgradnji poti med vozlišči. Na podlagi topoloških informacij, ki jih vsebuje vozlišče, se izračuna in izbere najprimernejšo (najoptimalnejšo) pot skozi omrežje. Na voljo je lahko več poti, pri izbiri pa gre večinoma za izbiro najkrajše poti. Najkrajša pot je tista pot, pri kateri je vsota povezav med izvornim in ponornim vozliščem najmanjša. Pri izračunu poti se lahko upoštevajo še ostale omejitve, kot sta na primer zakasnitev in pasovna širina. V tem primeru rezultat izbrane poti ni vedno najkrajša pot, temveč je to optimalna pot [4, 13].

Ko izberemo ustrezno pot med izvornim in ponornim vozliščem, se nanjo usmeri promet na osnovi posredovalne tabele. Posredovalna tabela se v nepovezavnem omrežju zgradi za vsako vozlišče posebej, neodvisno od ostalih vozlišč. Za izgradnjo tabel v povezavnem omrežju skrbijo signalizacijski protokoli. Učinkovito usmerjanje prometa prek poti omogočimo z mehanizmi za upravljanje prometa. Pomemben mehanizem za upravljanje prometa skozi omrežje je vzpostavitev dodatne poti, po kateri se posreduje del omrežnega prometa. S tem razbremenimo omrežje in izboljšamo parametre kakovosti storitev, kot so pasovna širina, zakasnitev, spreminjanje zakasnitve in izguba paketov.

4.1.1 Prometni inženiring v omrežju IP

V omrežju IP se s procesom usmerjanja določijo poti za omrežni promet, s pomočjo katerih nadzorujemo, koliko prometa prečka vsako povezavo. V omrežju IP se promet posreduje in usmerja na osnovi klasičnega usmerjevalnega protokola, kot je na primer OSPF. Usmerjevalni protokol izvaja usmerjevalni algoritem, ki izbere najkrajšo (najoptimalnejšo) pot v omrežju, po kateri se pošilja ves omrežni promet. Izbrana pot ni edina možna pot v omrežju. S stališča usmerjevalne metrike obstajajo tudi manj ugodne in nekoliko dražje poti, ki ostajajo neizkoriščene. Pri preobremenjenem omrežju se tako neenakomerno porazdelita omrežni promet in izraba virov, zato pride do preobremenjenega omrežja in v najslabšem primeru do izgube paketov. Problem je v strokovni literaturi [19] označen kot "fish-problem", ki ga

prikazuje slika 17. Tu pride do zasičenosti, ker se dve vrsti omrežnega prometa (rdeča in modra barva povezave) pošiljata po isti (najkrajši) poti. Najkrajša pot je tista, pri kateri je cena (vrednost) vseh povezav med izvorom in ponorom omrežnega prometa najmanjša. Na sliki 17, kjer se pošilja ves omrežni promet (rdeča in modra povezava), je vrednost najkrajše poti 4, saj pot poteka med dvema povezavama z vrednostjo 2. Obstaja še dodatna pot, po kateri se ne pošilja omrežni promet in je sestavljena iz treh povezav z vrednostjo 3, 2 in 1. Vsota vrednosti teh povezav je 5, zato to ni najkrajša poti.

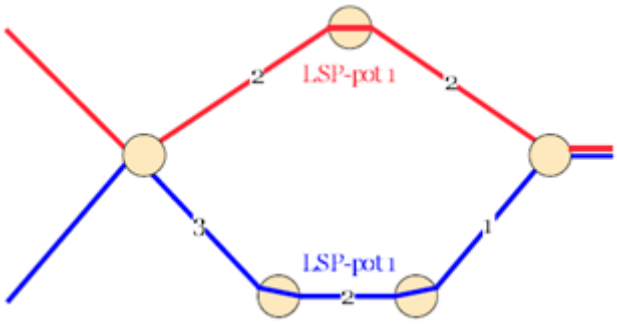


Slika 17: Promet v omrežju IP-"fish problem" [4]

Z obstoječimi nepovezavno usmerjenimi omrežji IP ne moremo učinkovito preusmeriti prometnih tokov na ostale poti, saj z njimi ni mogoče dinamično vzpostaviti nadomestnih (backup) poti ter tako optimalno izrabiti omrežnih virov. Prometni inženiring v omrežju IP je mogoč samo, če imajo vse poti enako ceno. Tega pogoja v realnem omrežju ne moremo izpolniti, zato se prometni inženiring v omrežju IP izvaja v povezavi s protokolom MPLS.

4.1.2 Prometni inženiring s protokolom MPLS

Prenos omrežnega prometa po različnih poteh dosežemo s prometnim inženiringom, ki nam ga omogoča protokol MPLS na osnovi povezavno usmerjenega pristopa. Z uporabo prometnega inženiringa vzpostavimo dodatno pot skozi omrežje MPLS. V primeru dveh vrst omrežnega prometa se eden izmed njiju preusmeri na dodatno vzpostavljeno pot (slika 18). V omrežju MPLS sta vzpostavljeni dve poti. Prva je obarvana z rdečo, druga pa z modro barvo. Zaradi pošiljanja omrežnega prometa po ločenih poteh se zmanjša možnost zasičenosti povezave in preobremenjenosti omrežja. Poleg tega se paketi prenašajo z manjšo zakasnitvijo, variacijo zakasnitve in izgubo, kar pripomore k boljši kakovosti storitev.



Slika 18: Omrežje MPLS s prometnim inženiringom [4]

S prometnim inženiringom upravljamo omrežni promet in tako učinkovito izrabljamo omrežne vire. Osnova za prometni inženiring v omrežjih MPLS je pot LSP. Ta se lahko vzpostavi na podlagi usmerjevalnih protokolov ali eksplicitno (ang. Explicitly Routed LSP – ER-LSP).

Če nimamo omejitev pri vzpostavljanju poti, se pot LSP vzpostavi od vozlišča do vozlišča s pomočjo usmerjevalnega protokola (npr. protokola OSPF). V tem primeru vsak usmerjevalnik MPLS (usmerjevalnik LSR) na osnovi usmerjevalne topološke baze določi naslednji usmerjevalnik. V skladu s tem pošlje naslednjemu usmerjevalniku LSR (naslednjemu hopu) zahtevo za dodelitev labele. Z dodelitvijo labele na vsakem usmerjevalniku LSR je vzpostavljena pot LSP. Torej je pot LSP v omrežju MPLS enaka najkrajši poti v omrežju IP.

Eksplicitna pot

Pot, pri kateri upoštevamo omejitve pri vzpostavljanju, se imenuje eksplicitna pot ER-LSP. Omejitve lahko določi operater omrežja ali pa se izračunajo z uporabo algoritma za upravljanje omrežij, ki je neodvisen od privzetih usmerjevalnih protokolov IP. Celotna pot LSP se določi s "Setup" sporočilom, ki se posreduje po zahtevani poti. Usmerjevalniki LSR v tem primeru ne uporabljajo usmerjevalnih informacij, ampak zahtevo za dodelitev labele pošljejo tistemu naslednjemu "hopu", ki je določen v "Setup" sporočilu [4].

Eksplicitna pot je osnova prometnega inženiringa. Določi se v izvornem vozlišču in je neodvisna od usmerjanja po najkrajši poti. Eksplicitna pot predstavlja zaporedje vozlišč, ki so lahko izbrana s strani operaterja oziroma algoritma. Operater določi eksplicitno pot na podlagi omejitev, kot so pasovna širina in uporabnost ter neuporabnost poti. Eksplicitna pot se lahko izračuna s pomočjo algoritma na podlagi topologije in uporabniških zahtev.

Protokol MPLS omogoča še dodatno fleksibilnost za upravljanje eksplicitnih poti. Le-te so lahko določene natančno (ang. strict) ali ohlapno (ang. loose). V primeru striktno določenih eksplicitnih poti operater natančno specificira vso pot (vozlišča), medtem ko v primeru ohlapno določenih eksplicitnih poti operater natančno določi samo del poti, preostali del pa prepusti omrežnim usmerjevalnim protokolom. S tem se pridobi fleksibilnost. Slaba stran ohlapno določenih eksplicitnih poti so potencialni problemi s stabilnostjo in manjša preglednost omrežja.

4.2 Kakovost storitev

Kakovost storitev (ang. Quality of Service – QoS) definiramo kot nabor pravil in parametrov za prenos podatkov med aplikacijami in omrežji [3]. Pravila prenosa podatkov se lahko dinamično spreminjajo glede na trenutne zahteve, ki izhajajo iz aplikacij in podatkovnih tokov le-teh. Za vsako aplikacijo veljajo drugačne zahteve prenosa podatkov. Aplikacije (npr. video, zvok), ki prenašajo podatke v realnem času, morajo ustrezati bolj strogim zahtevam kot ostale aplikacije (npr. elektronska pošta). Optimalen prenos podatkov lahko zagotovi le omrežje, ki ponuja kakovost storitev, ki se čim bolj prilagaja zahtevam in prometnim pretokom aplikacij.

Osnovna verzija omrežja IP ni omogočala nikakršnih zagotovil o kakovosti storitev. Uporabnikom je omogočala samo povezljivost za prenos podatkov med pošiljateljem in prejemnikom. V omrežju IP se prenos podatkov izvaja po principu "najboljših zmožnosti" (ang. best effort). To pomeni, da se ne ločuje med podatki različnih aplikacij in se vse

obravnavajo enako. Tako se aplikacija, kot je elektronska pošta, obravnava enako kot aplikacija za prenos glasovnega prometa, čeprav imata popolnoma različne zahteve glede prenosa podatkov. V nizko obremenjenih omrežjih to ni problem, saj vsaka aplikacija lahko dobi svoj delež omrežnega vira. Problem se pojavi z večanjem obremenjenosti omrežja, ker se razmere za vse aplikacije poslabšajo enako. Za aplikacijo, kot je elektronska pošta, to ni ovira, saj ne izkazuje posebnih zahtev glede zakasnitve in drugih parametrov kakovosti storitev. Večja težava nastopi pri prenosu govornega prometa preko omrežja IP, saj zahteva precej več. Poleg stalne in rezervirane pasovne širine zahteva tudi nizko in čimmanj spremenljivo zakasnitev. Za omrežje IP so to precej hude zahteve, ki jih lahko zagotovimo z vpeljavo mehanizma za zagotavljanje kakovosti. Uveljavila sta se dva mehanizma. Prvi se imenuje model integriranih storitev (ang. Integrated Services – IS), drugi pa model diferenciranih storitev (ang. Differentiated Services – DS).

Integrirane storitve

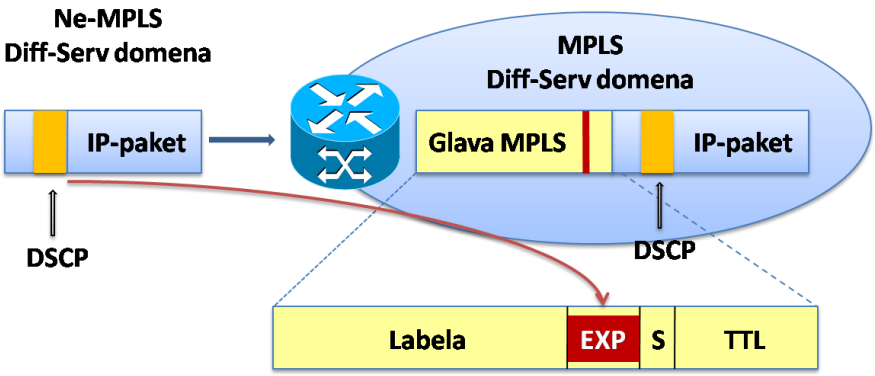
Integrirane storitve v omrežje IP uvedejo tri razrede storitev. Poleg razreda najboljših zmožnosti sta uvedena še razreda zagotovljenih storitev (ang. Guaranteed Services) in storitev nadzorovanega bremena (ang. Controlled Load Services). Razred zagotovljenih storitev je namenjen aplikacijam s strogimi zahtevami glede prenosnih parametrov, kot sta pasovna širina in zakasnitev, zato je zelo primeren za vse aplikacije, ki zahtevajo prenos podatkov v realnem času. Razred storitev nadzorovanega bremena se uporablja v primerih, ko aplikacije nimajo postavljenih tako strogih kriterijev, kot pri zagotovljenih storitvah. Ta razred je primeren predvsem za aplikacije, ki zahtevajo boljše lastnosti prenosa podatkov, kot jih lahko nudi prenos po najboljših zmožnostih [3].

Diferencirane storitve

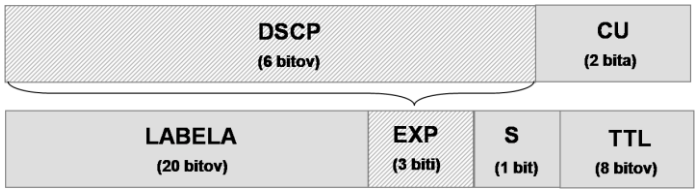
Zaradi omejitev modela integriranih storitev in zahtevne izvedbe le-tega se je v omrežju IP uveljavil model diferenciranih storitev. V omrežju IP se z modelom diferenciranih storitev ne spremlja vsaka aplikacija posebej, temveč se spremlja skupina (razred) aplikacij s podobnimi zahtevami. Takšna skupina se imenuje vedenjski skupek (ang. Behavior Aggregate – BA) in jo uvrstimo v ustrezni razred glede na zahtevano kakovost. Za nastavitev ustreznega razreda se uporablja 6 bitov v glavi IP-paketa, ki se imenujejo DSCP (ang. DiffServ Code Point). Hrbtenica omrežja na osnovi določenega razreda zagotavlja ustrezno kakovost. Kakovost se zagotavlja na vsaki točki (usmerjevalniku) posebej (ang. Per-Hop Behaviour – PHB), to pa je slabost modela diferenciranih storitev. Upošteva se samo stanje v trenutni točki, ne pa stanje celotnega omrežja.

Z nadgradnjo protokola MPLS v omrežju IP se ne definira novi model za zagotavljanje kakovosti storitev. V omrežju MPLS se zagotavlja kakovost storitev na osnovi uveljavljenega modela diferenciranih storitev. Kot smo spoznali, se za prenos IP-paketov skozi omrežje MPLS uporabljajo labele. Zaradi IP-paketov, opremljenih z labelo, je polje DSCP v glavi IP-paketa za usmerjevalnik nevidno. Usmerjevalnik pri razvrščanju paketov v ustrezne razrede tako namesto polja DSCP v glavi IP-paketa uporablja eksperimentalne bite, ki so v labeli pod poljem EXP. Polje EXP je dolgo 3 bite, zato se lahko zagotovi le do 8 razredov kakovosti storitev. Polje DSCP, ki se vpelje z modelom diferenciranih storitev, omogoča bistveno več razredov, saj je dolgo 6 bitov. Če želimo v omrežju MPLS zagotoviti kakovost storitev na osnovi modela diferenciranih storitev, je treba rešiti problem preslikave med poljem DSCP in poljem EXP, kot je splošno prikazano na sliki 19 in podrobno na sliki 20.

Problem se rešuje na dva načina: z uporabo poti E-LSP (ang. EXP-inferred-PSC[3] LSP – LSP z določitvijo PHB na podlagi polja EXP) in poti L-LSP (ang. Label-only-inferred-PSC LSP – LSP z določitvijo PHB samo na podlagi vrednosti labele).



Slika 19: Splošen prikaz preslikave med poljem DSCP in poljem EXP [7]



Slika 20: Podroben prikaz preslikave med poljem DSCP in EXP

Pot E-LSP

V primeru poti E-LSP se polje DSCP prilagodi polju EXP, tako da se preslikajo samo spodnji biti polja DSCP v polje EXP. Ostali biti se zavržejo. Paketi opremljeni z labelo omogočajo do 8 razredov kakovosti storitev. Dobra stran uporabe poti E-LSP je majhna poraba prostora label. Slaba stran je majhna fleksibilnost pri določanju poti in omejenost na osem razredov kakovosti storitev.

Pot L-LSP

Pot L-LSP se uporabi, če želimo izkoristiti več razredov storitev, kot jih ponuja polje EXP. S poljem EXP v labeli lahko zagotovimo do 8 razredov storitev, s poljem DSCP pa do 64 razredov storitev. Razrede kakovosti storitev diferenciranega modela omogočimo z razširitvijo polja EXP v labeli. Če to ni mogoče, se uporabijo ločene labele za vsak razred kakovosti storitev. Prednost uporabe poti L-LSP je v podpori velikega števila razredov kakovosti storitev in v fleksibilnem določanju poti. Pomanjkljivost je velika poraba label.

[3] PSC (ang PHB Scheduling Class –razporeditveni razred PHB) je eden ali več PHB-jev, ki se uporabljajo za vedenjske skupke in si delijo omejitve vrstnega reda prenosa paketov (RFC 3270).

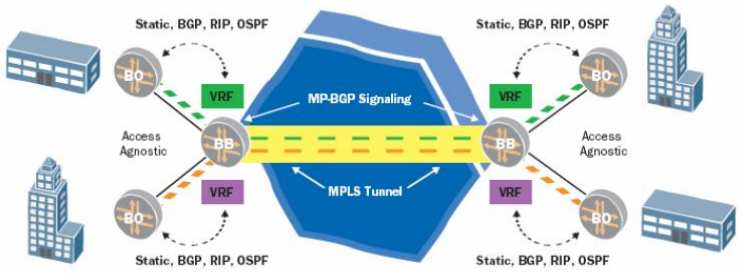
4.3 Navidezna zasebna omrežja

Navidezno zasebno omrežje (ang. Virtual Private Network – VPN) je omrežje, ki uporablja skupno omrežno infrastrukturo, kot je internet, za povezavo dveh oddaljenih omrežij. Osnova za navidezna zasebna omrežja je tunelska povezava, ki se vzpostavi med večjimi področji dveh lokacij. S tunelsko povezavo se zagotovi varna (kriptirana) in zanesljiva povezava. Preko tunelske povezave se prenašajo paketi različnih protokolov (IP in TCP), ki so vgrajeni v druge prenosne protokole. Eden izmed takih protokolov je protokol IPSec (ang. Internet Protocol Security). Protokol IPSec je kriptirani protokol in se nahaja na omrežnem nivoju.

MPLS VPN je način uporabe tehnologije MPLS za graditev navideznih zasebnih omrežij. Za ta namen je MPLS zelo primerna tehnologija, saj omogoča izolacijo in razločevanje prometa praktično brez dodatne režije. Omrežja MPLS VPN omogočajo varne povezave in relativno preprosto konfiguracijo. Zaradi popularnosti se uporabljajo tako za internet kot tudi za ekstranet. Cilj MPLS VPN tehnologije je zgraditi omrežje, ki naj deluje kot podaljšek zasebne korporativne omrežne infrastrukture preko omrežja nekega ponudnika storitev. Na ta način lahko povežemo geografsko razpršene lokacije, kot so na primer podružnice podjetij. Tako povežemo podružnice podjetij v skupno, za uporabnika transparentno okolje [20].

4.3.1 Navidezna zasebna omrežja MPLS na tretji plasti

Pri tej vrsti navideznih zasebnih omrežij poteka transport prometa skozi omrežje preko tunelov MPLS s pomočjo signalizacijskega protokola MP-BGP (ang. Multi Protocol BGP).

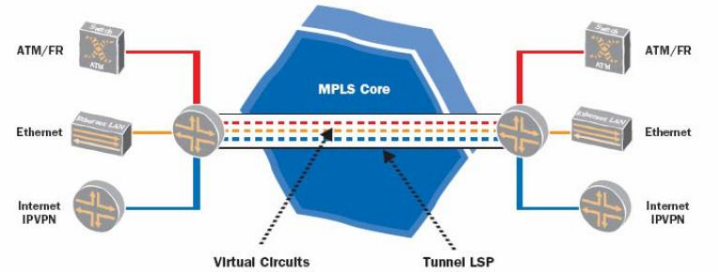


Slika 21: Navidezno zasebno omrežje MPLS na tretji plast [18]

Na sliki 21 je hrbtenični usmerjevalnik MPLS (ang. Backbone Router – BB) in usmerjevalnik pri stranki (ang. Branch Office – BO), ki pa ne deluje v načinu MPLS. To je najpogostejši način uporabe navideznih zasebnih omrežij MPLS. Obstaja tudi možnost, da se MPLS razširi do strankine lokacije in se tako vzpostavi od ene končne točke do druge. Vsak hrbtenični usmerjevalnik MPLS vsebuje usmerjevalno instanco, t.i. predajo navideznih smeri (ang. Virtual Route Forwarding – VRF). Na ta način je hkrati v enem usmerjevalniku prisotnih več neodvisnih usmerjevalnih tabel. Ker so medsebojno neodvisne, s tem dosežemo, da lahko različni uporabniki omrežja MPLS uporabijo iste zasebne naslove IP za svoje navidezno zasebno omrežje, ne da bi s tem vplivali drug na drugega. S pomočjo predaje navideznih smeri in izolacije prometa tako kreiramo navidezna omrežja. Prednost navideznih zasebnih omrežij na tretji plasti je, da so standardizirana in dokaj enostavna. Ta tip navideznih zasebnih omrežij podpira širok nabor tipov dostopa in več topologij hrbteničnega omrežja. Taka rešitev je bolj razširljiva in cenejša od klasičnih omrežij ATM in Frame Relay ali navideznih zasebnih omrežij z uporabo IPsec-a. Ta pristop podpira tudi mehanizme za kakovost storitev [20].

4.3.2 Navidezna zasebna omrežja MPLS na drugi plasti

Navidezna zasebna omrežja na drugi plasti (slika 22) so zelo razširjena in temeljijo na ATM in Frame Relay tehnologiji. Pri takih navideznih zasebnih omrežjih je najpomembnejša vzpostavitev tunelov oziroma poti LSP.



Slika 22: Navidezno zasebno omrežje MPLS na drugi plast [18]

Za nadzorni protokol se uporabljata protokola LDP in BGP, s katerima se vzpostavljajo navidezni vodi. Prednost navideznih zasebnih omrežij druge plasti pred navideznimi zasebnimi omrežji tretjega sloja je, da podpirajo velik nabor različnih enkapsulacij, kot sta Ethernet in ATM. Ena izmed slabosti je potreba po individualni konfiguraciji vsakega navideznega voda. Zaradi tega navidezna zasebna omrežja druge plasti niso najbolj razširljiva oziroma skalabilna [20].

5 Primerjava omrežja IP in omrežja MPLS

V tabeli 1 je opisana primerjava omrežja IP brez protokola MPLS in z njim. Kot se vidi iz tabele, se je omrežje MPLS izkazalo za boljše v vseh kriterijih.

Št.	Kriteriji	Omrežje IP	Omrežje MPLS
1.	Prenos IP-paketov	na podlagi ciljnega naslova, z usmerjanjem	na podlagi label, s preklapljanjem
2.	Usmerjevalne tabele	velike in kompleksne	majhne in preproste
3.	Hitrost usmerjanja IP-paketov	nizka	visoka
4.	Prilagodljivost s prenosnimi tehnologijami druge plasti OSI referenčnega modela	ni prilagodljiv	izredno prilagodljiv
5.	Kakovosti storitev (QoS)	zagotovljena	zagotovljena
6.	Možnost uporabe prometnega inženiringa	manjša	večja

Tabela 1: Primerjava omrežja IP in omrežja MPLS

Omrežje IP s protokolom MPLS je izredno prilagodljivo in zanesljivo, zato je uporaba protokola velika pridobitev za ponudnike omrežnih storitev. V omrežju MPLS se prenos IP-paketov izvaja s pomočjo label, ki so po zgradbi majhne in preproste. Labele zagotavljajo enostavno in učinkovito rešitev za omrežje MPLS, ker se z njimi izognemo kompleksni analizi glave IP-paketa na vsakem vozlišču. Labela se dodeli vsakemu IP-paketu enkrat, in sicer na začetku poti pošiljanja IP-paketa skozi omrežje. IP-paketi se skozi omrežje posredujejo s hitrim preklapljanjem na osnovi label, in ne z usmerjanjem na osnovi ciljnega naslova IP. Tako ni treba v vsakem vozlišču pregledati usmerjevalne tabele in ugotavljati, katero bo naslednje vozlišče, kamor bo IP-paket usmerjen. Zaradi tega se zmanjša velikost usmerjevalnih tabel in s tem tudi kompleksnost strojne opreme, ki posreduje IP-pakete skozi omrežje. Enostavna strojna oprema nam omogoča hitrejše procesiranje IP-paketov in s tem visoko hitrost prenosa IP-paketov.

Omrežje IP ni prilagodljivo za potrebe prenosnih tehnologij druge plasti, zato je bil vpeljan protokol MPLS, ki deluje neodvisno od ostalih protokolov in prenosnih tehnologij. V omrežju IP so omejitve glede hitrosti prenosa podatkov. Pri povečanju števila uporabnikov v omrežju IP in posledično večjega števila prenesenih podatkov lahko dobimo manj zanesljivo in počasno delovanje omrežja. Rešitev za strmo povečanje števila uporabnikov je hitro posredovanje IP-paketov na osnovi label. Tako postane omrežje MPLS bolj skalabilno, to pomeni, da se je neko omrežje zmožno prilagoditi na povečanje števila uporabnikov. Večje število uporabnikov pomeni večje število naslovov IP, ki jih lahko preprosto povežemo z eno ali z nekaj labelami [21].

V omrežju IP in omrežju MPLS se zagotovi kakovost storitev na osnovi modela diferenciranih storitev. V omrežju IP se IP-paketi razvrščajo glede na polje DSCP v glavi IP-paketa, v omrežju MPLS pa se razvrščajo glede na polje EXP, ki se nahaja v labeli.

Uporaba prometnega inženiringa je v omrežju MPLS večja, saj se lahko na enostaven način, s pomočjo signalizacijskih protokolov, vzpostavijo poti skozi omrežje. Tako lahko vzpostavimo dodatne poti skozi omrežje, s katerimi razbremenimo omrežje in izboljšamo kakovost storitev.

6 Vrednotenje računalniških omrežij

Pri vrednotenju zmogljivosti računalniških omrežij se uporabljajo različne tehnike, kot so meritve, analitično modeliranje in simulacija računalniških omrežij.

Meritve so najučinkovitejša metoda, ker se z njimi zagotavlja velika natančnost pri pridobivanju podatkov. Slaba stran meritev sta draga merilna oprema in zamudno odpravljanje napak. Problem vrednotenja z meritvami se pojavi pri vzpostavitvi novih sistemov oziroma omrežij, ko meritve še ni mogoče izvesti.

Pristop z analitičnim modeliranjem temelji na izdelavi matematičnega modela, preko katerega se rešujejo različne vrste sistemov enačb. Analitični model predstavlja le posplošen, abstrakten in poenostavljen opis realnega omrežja. Pri velikih omrežjih so rezultati, dobljeni s pristopom analitičnega modeliranja, le približek realnih rezultatov.

Simulacija se uporablja pri modeliranju sistemov na področju inženirskih raziskav, poslovnih analiz, načrtovanj proizvodnje in bioloških znanstvenih raziskav. Po mnenju Shannona [31] je simulacija proces načrtovanja modela realnega sistem in izvajanja poskusov nad tem modelom.

Izbira tehnike za vrednotenje je odvisna od kriterijev, ki jih podaja tabela 2.

Kriterij	Analitično modeliranje	Simulacija	Meritve
Obdobje v katerem je sistem	kadarkoli	kadarkoli	če je model že narejen, lahko na njem opravljamo meritve
Čas za pridobitev rezultatov	majhen	srednji	različen (ponavadi gre pri testiranju vse narobe)
Orodja, ki so na voljo	oblikovalske sposobnosti	računalniški programi	naprave, instrumenti
Točnost rezultatov	majhna	srednja	odvisno od izbire parametrov
Ocenjevanje spremembe parametrov	lahko	srednje	težko
Stroški	majhni	srednji	veliki
Vpliv na prodajo rezultatov analize	majhen	srednji	velik

Tabela 2: Kriteriji za izbiro tehnike vrednotenja [29]

6.1 Simulacija računalniškega omrežja

Simulacijo računalniškega omrežja izvajamo, da razumemo delovanje sistema in ovrednotimo različne strategije operacij sistema. Simulacije temeljijo na dinamičnih sistemih, kar je zelo uporabno na področju izvajanja simulacij dinamičnih računalniških omrežij.

Simulacija je lahko mišljena kot proces omrežnega toka entitet (paketov). Pri premikanju entitet skozi sistem le-te medsebojno vplivajo na ostale entitete, združujejo določene aktivnosti, prožijo dogodke, povzročijo spremembo in čakajo na omrežne vire.

Komponente za simulacijo so naslednje [28]:

- **Entitete** so objekti, ki povzročijo spremembe v stanju sistema in v simulaciji medsebojno vplivajo druga na drugo. V simulaciji računalniških omrežij so entitete na primer paketi.
- **Resursi** so del kompleksnih sistemov in so porazdeljeni med določeno množico vozlišč. Primer resursov sta pasovna širina in število strežnikov.
- **Dogodki** predstavljajo spremembo stanja sistema. Sprememba stanja sistema se pojavi ob prihodu zahteve v sistem, strežno vrsto itd. Dogodki se prožijo, ko se vozlišču dodajo določene aktivnosti, na primer generiranje zahtev, ki se pošiljajo v sistem.
- **Razvrščevalnik** vzdržuje seznam dogodkov, ki se sprožijo v času izvajanja simulacije. Skozi simulacijo razvrščevalnik izvaja simulacijsko uro in ustvari ter izvede dogodke.
- **Globalne spremenljivke** so v simulaciji sistema dosegljive preko funkcije ali entitete in ohranjajo osnovne sledi nekaterih pogostih vrednosti simulacije. Spremenljivke so lahko dolžina strežne vrste ali celotno število poslanih paketov.
- **Naključni generatorji števil** so potrebni za vpeljavo naključnosti v simulacijski model. V simulaciji računalniških omrežij se naključni proces pojavlja pri procesu prihoda, čakanja in strežbe paketa.
- **Zbiratelj statistik** za zbiranje podatkov, ki se generirajo v procesu simulacije.

Poznamo več vrst simulacij, ki se razlikujejo po vrsti stanja sistema in vrsti časa. Stanje sistema je lahko definirano diskretno ali zvezno. Primer diskretnega stanja sistema je število zahtev v čakalni vrsti. Število zahtev je lahko samo pozitivno in celo število. Primer zveznega stanja sistema je izmerjena temperatura s termometrom, ki ima lahko poljubno realno vrednost. Stanje sistema se lahko spreminja v diskretnem ali zveznem času. Primer stanja sistema, ki se spreminja v diskretnem času, bi bila simulacija števila obiskovalcev predavanj vsak ponedeljek dopoldan ob 8.00. Zvezni čas se uporabi pri obravnavi časa, ki ga zahteva prebije v čakalni vrsti. Simulacija računalniških sistemov običajno poteka v zveznem časovnem prostoru z diskretnimi stanji sistema [23].

V našem primeru bomo za implementacijo simulacijskega modela uporabili diskretno dogodkovno simulacijo. Simulacija diskretnih dogodkov omogoča določanje stanj sistema v odvisnosti od časa, zbiranje podatkov in ustrezno statistično analizo. Pri diskretni dogodkovni simulaciji bomo stanje sistema opisali s pomočjo diskretnih dogodkov. Izvajanje diskretno dogodkovne simulacije bo potekalo po kronološkem vrstnem redu. Pri analizi rezultatov simulacijskega modela gre za časovno zvezno simulacijo, saj se stanje spremenljivk sistema (npr. čas, ko je paket v strežni vrsti) spreminja v zveznem času.

6.2 Omrežni simulatorji za simulacijo računalniških omrežij

Omrežni simulator je program, ki imitira delovanje realnega računalniškega omrežja. Z omrežnim simulatorjem napovemo obnašanje omrežja brez dejanske postavitve realnega omrežja. Omrežni simulatorji poskušajo modelirati realna omrežja. Osnovna ideja omrežnega simulatorja je simuliranje sistema s spreminjanjem značilnosti sistema in analiza ustreznih rezultatov. Proces spreminjanja modela je relativno poceni v primerjavi z izgradnjo dejanskega realnega omrežja.

Omrežni simulator ne omogoča simuliranja vseh podrobnosti omrežij. S pravilno simulacijo omrežnega simulatorja se lahko zelo dobro približamo delovanju realnega omrežja. S simulacijo omrežnega simulatorja lahko preizkušamo delovanje omrežij in spremljamo vpliv sprememb na delovanje.

Poznamo komercialne in odprtokodne omrežne simulatorje. Komercialni omrežni simulatorji ne zagotavljajo dostopa do izvorne kode programa. Vsi uporabniki morajo za uporabo programa kupiti licenco, ki je zelo draga. Odprtokodni omrežni simulatorji so javno dostopni, kar omogoča dostop do izvorne kode programa in s tem njegovo nadgradnjo. Slabost odprtokodnih simulatorjev je, da dokumentacija ni sistematična in dovršena. Najbolj znana odprtokodna simulatorja sta OMNET++ in NS-2.

OPNET je komercialni simulator, ki se uporablja za raziskovanje in razvijanje komunikacijskih omrežij, naprav, protokolov in aplikacij. OPNET daje uporabnikom široko vizualno in grafično podporo. Z grafičnim uporabniškim vmesnikom lahko preprosto zgradimo omrežno topologijo in nastavimo parametre za simulacijo omrežij. OPNET uporablja objektni programski jezik C++ in zagotavlja virtualno okolje za modeliranje, analiziranje in predpostavljanje zmogljivosti omrežij.

OMNeT++ je objektno usmerjen diskretni simulator z grafičnim uporabniškim vmesnikom. Primarna uporaba je simulacija računalniških omrežij na področju telekomunikacijskih sistemov, omrežij z množično strežbo in arhitekture strojne opreme. Z njim se omogoča komponentna arhitektura za simulacijske modele. Modeli so sestavljeni iz komponent (modelov), ki se uporabijo in združijo na različne načine. OMNET++ ponuja samo osnovne mehanizme in orodja za izvajanje simulacij. Vse ostale komponente, ki jih potrebujemo pri izvajanju simulacij, je treba vpeljati v orodje.

NS-2 je eden izmed najbolj razširjenih omrežnih simulatorjev. Uporablja se v raziskavah računalniških omrežij. Razvit je bil leta 1989 na osnovi simulatorja *REAL network simulator*. NS-2 je simulator časovnih diskretnih dogodkov, kjer je potek časa odvisen od časovnega usklajevanja dogodkov, ki jih upravlja razvrščevalnik dogodkov. Napisan je bil predvsem zaradi simulacije omrežij IP, uporaben pa je tudi za študije drugih vključenih omrežij. Glavna slabost omrežnega simulatorja sta slabo dokumentirana izvorna koda in težavno vrednotenje majhnih idej, saj je treba dobro poznati vso strukturo programov.

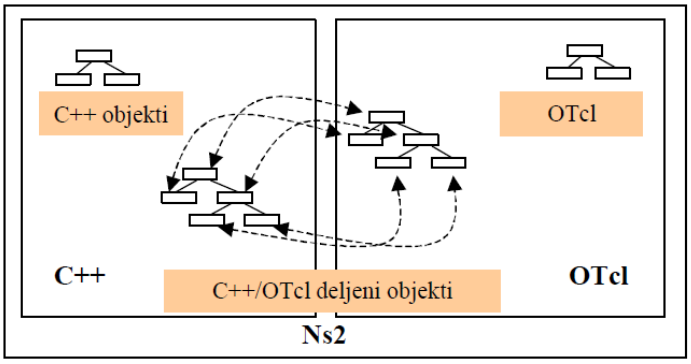
7 Omrežni simulator NS-2

7.1 Zgradba NS-2

Omrežni simulator NS-2 omogoča simulacijo časovnih diskretnih dogodkov. Podpira različne omrežne protokole (npr. TCP, UDP), načine usmerjanja (multicast in unicast), več vrst povezav (žične, brezžične in satelitske) in vrst prometa (CBR, FTP in telnet). Omrežni simulator temelji na dualni zasnovi programskih jezikov C++ in na preprostem skriptnem jeziku OTcl (ang. Object-Oriented Tool Command Language). Programski jezik OTcl je interpreter, zato ne potrebujemo prevajanja ob vsaki spremembi v izvornem programu. Uporablja preprosto sintakso in omogoča hiter razvoj simulacijskih modelov.

Programski jezik OTcl se uporablja za:
- hitro načrtovanje topologije omrežij,
- izdelavo sprememb omrežnih scenarijev,
- raziskovanje zmogljivosti omrežij.

Jezik OTcl je počasen pri izvajanju simulacij, zato se v ta namen uporabi jezik C++, s katerim se omogoča hitro izvajanje simulacij. Zaradi hitrega izvedbenega časa programov, napisanih v jeziku C++, je le-ta primeren za natančen opis in izvajanje protokolov. Jezik C++ je neprimeren za spremembe omrežnih scenarijev, ker je ob vsaki spremembi treba prevesti program. To zahteva določen čas.



Slika 23: Dualnost C++ in OTcl [10]

Dobra lastnost dveh programskih jezikov (slika 23) je kompromis med sestavljanjem scenarija in hitrostjo izvajanja. Slabost je v razumevanju dveh jezikov in odpravljanju napak v dveh programskih jezikih.

Za potrebe izvajanja simulacij računalniškega omrežja bomo uporabili omrežni simulator NS-2, ker omogoča:
- preprosto nastavitev simulacije,
- hitro izvajanje simulacije,
- podporo večini protokolov,
- odprto kodo za spreminjanje.

7.2 Modul MPLS v NS-2

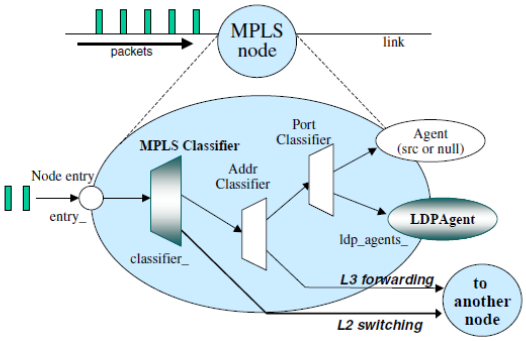
Modul MPLS je eden izmed modulov omrežnega simulatorja NS-2. Prvotno je bil zasnovan kot samostojen modul in se je imenoval MNS [12] (ang. MPLS Network Simulator). Modul MPLS je bil zgrajen z razširitvijo omrežnega simulatorja NS-2, ki je osnova za simulacijo omrežja IP. Modul MPLS omogoča izvajanje protokola MPLS. Z njim simuliramo različne aplikacije MPLS brez izgradnje realnega omrežja MPLS.

V modulu MPLS so omogočene naslednje funkcije:

- **preklapljanje paketov na osnovi label** – operacije z labelami (push, swap, pop), zmanjševanje TTL in odstranitev labele na predzadnjem vozlišču z operacijo PHP (ang. Penultimate Hop Popping),
- **distribucijo label** s protokolom LDP,
- **protokol CR-LDP** – podpora eksplicitnim potem (ER-LSP).

7.2.1 Arhitektura vozlišča MPLS

V omrežnem simulatorju NS-2 je vozlišče IP sestavljeno iz agentov in razvrščevalnikov (ang. Classifiers). Agent je objekt pošiljatelja oziroma prejemnika protokola (npr. TCP, UDP), razvrščevalnik pa je objekt, ki razvršča pakete. Vozlišče MPLS (ang. MPLS node) je razširjeno vozlišče IP z dodanim razvrščevalnikom MPLS (ang. MPLS Classifier) in agentom LDP (ang. LDP agent), ki je objekt protokola LDP. Na sliki 24 je prikazana arhitektura vozlišča MPLS v omrežnem simulatorju.



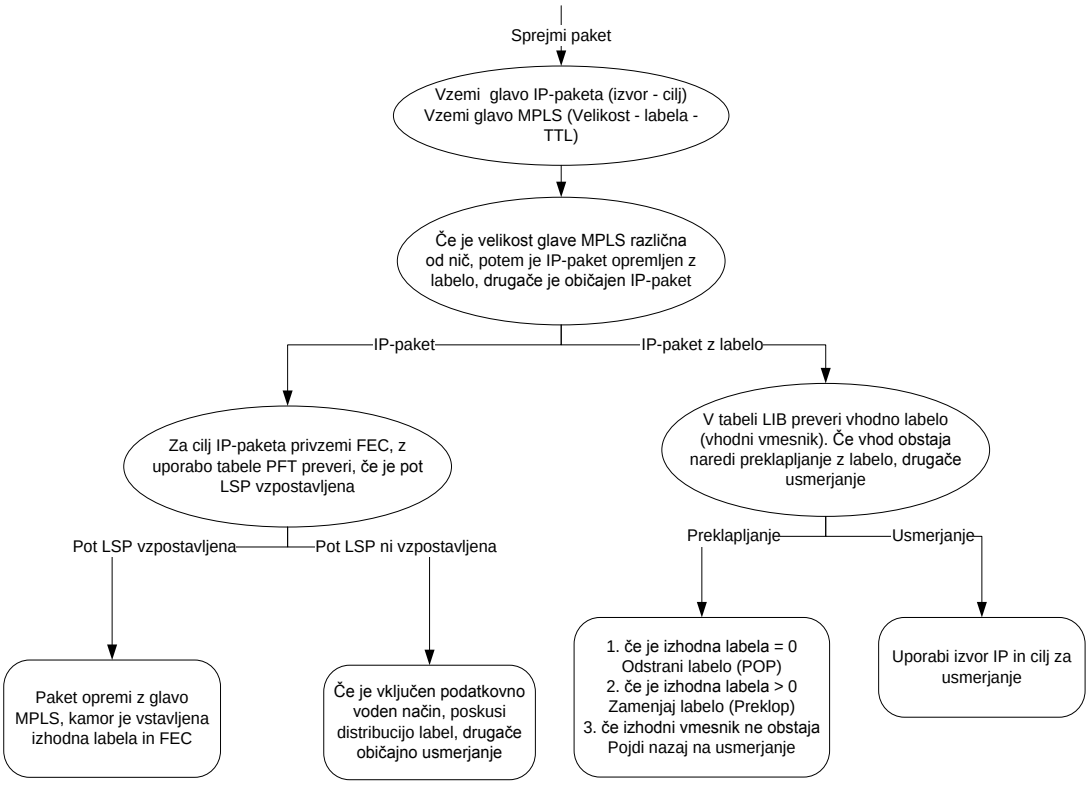
Slika 24: Vozlišče MPLS [12]

Po sprejetju paketa v vozlišču MPLS (ang. MPLS node) razvrščevalnik razvrsti prejete pakete glede na to, ali so opremljeni z labelo ali ne. Če so sprejeti paketi opremljeni z labelo, razvrščevalnik izvede operacijo z labelo in preklapljanje na povezavni plasti (ang. Layer 2 switching). Preklapljanje na povezavni plasti pomeni pošiljanje paketa, opremljenega z labelo, direktno na naslednje vozlišče. Če paket ni opremljen z labelo, ga razvrščevalnik MPLS pošlje do razvrščevalnika naslovov (ang. Addr Classifier). Razvrščevalnik naslovov izvede posredovanje paketa (ang. L3 forwarding) na osnovi ciljnega naslova. Paket se posreduje do naslednjega vozlišča oziroma do razvrščevalnika vhodov (ang. Port Classifier), če je paket na ciljnem vozlišču. Naloga razvrščevalnika vhodov je izbira agenta, ki sprejme paket na koncu njegove poti. Za izmenjavo sporočil LDP mora razvrščevalnik vhodov izbrati agenta LDP.

Vozlišče MPLS vsebuje tri tabele, ki so potrebne za vzpostavljanje poti LSP in distribucijo label:

- **Tabela LIB** vsebuje informacijo za vzpostavitev poti LSP. V tabeli LIB so podatki o vhodnem vmesniku (vozlišču) in vhodni labeli ter podatki o izhodnem vmesniku in izhodni labeli. Uporablja se v procesu dodeljevanja in zamenjave label. Vsebuje polje za kazalec LIBptr, kamor kažejo ostale tabele.
- **Tabela PFT** (ang. Partial Forwarding Table) je podmnožica usmerjevalne tabele in se uporabi v primeru, ko paket ni opremljen z labelo. V tabeli so polja: FEC, PHB in kazalec LIBptr na tabelo LIB. Če kazalec LIBptr v usmerjevalni tabeli kaže na vrednost nič, vozlišče MPLS posreduje pakete na podlagi ciljnega naslova, sicer vozlišče MPLS dodeli paketu labelo.
- **Tabela ERB** (ang. Explicit Routing Base) se ne uporabi pri procesu posredovanja paketov, temveč vsebuje informacijo za vzpostavitev eksplicitne poti. Polja v tabeli so: LSPid, FEC in LIBptr.

Celotni proces obdelave paketa v vozlišču MPLS prikazuje diagram na sliki 25.



Slika 25: Celotni diagram obdelave paketa v vozlišču MPLS

7.2.2 Modeli za razširjanje label v modulu MPLS

Osnovni načini za razširjanje label v modulu MPLS so:

- kontrolno voden način,
- podatkovno voden način,
- eksplicitno usmerjanje.

Kontrolno vodeni način temelji na razširjanju label med vsemi agenti LDP, čeprav v času izvajanja simulacije ni prisotnih nobenih paketov za prenos. Poti LSP se vzpostavijo za vsako ciljno vozlišče (FEC) s pošiljanjem sporočil preslikave (ang. mapping) med vsemi agenti LDP. Na koncu se zapolnijo vse tabele LIB in se dodelijo različne poti za vsako ciljno vozlišče.

Podatkovno vodeni način razširja labele med vozlišči, ki prenašajo pakete. Če paket prispe v vozlišče, se pošlje zahteva do ciljnega vozlišča, s katero se zgradi pot LSP. Pot LSP se zgradi samo za ciljna vozlišča, ki predstavljajo ponor paketov. Dokler pot LSP ni zgrajena, se paketi posredujejo na osnovi ciljnega naslova, nato se posredujejo s preklapljanjem na osnovi label.

V eksplicitnem usmerjanju label so poti LSP zgrajene na preprost način. Uporabnik mora podati vozlišča eksplicitne poti, po kateri bo potoval paket. Preko eksplicitne poti se prenaša samo sporočilo preslikave, s katerim se ustvari pot LSP za ciljno vozlišče.

7.2.3 Ukazi za simulacijo omrežja MPLS

Za simulacijo omrežja IP s protokolom MPLS v omrežnem simulatorju NS-2 so na voljo ukazi v programskem jeziku OTcl, ki jih omogoča modul MPLS. Z ukazi zgradimo topologijo omrežja MPLS in določimo potek dogajanja v omrežju. V omrežju MPLS je treba najprej določiti vozlišča s protokolom MPLS. Ko izberemo vozlišča, nad njimi omogočimo protokol LDP z uporabo agenta LDP. S tem omogočimo ključni funkciji, kot sta preklapljanje paketov na osnovi label in vzpostavitev eksplicitnih poti.

1. Ukazi za izgradnjo vozlišča MPLS (LSR0):

```
$ns node-config -MPLS ON
set LSR0  [$ns node]
$ns node-config -MPLS OFF
```

Vozlišča, nad katerimi želimo omogočiti protokol MPLS, so podana med ukazom "*node-config-MPLS ON*" in "*node-config -MPLS OFF*". S prvim ukazom vklopimo protokol MPLS, z drugim ukazom ga izklopimo. Vozlišče ustvarimo z ukazom "*set LSR0 [$ns node]*". Primer vklopa protokola MPLS izvedemo nad vozliščem LSR0.

2. Nastavitev agentov LDP na vozlišču MPLS (prikaz za n vozlišč, katerega imena so LSRi, pri čemer je i število):

```
for {set i 0} {$i < n} {incr i} {
        set a LSR$i
        for {set j [expr $i+1]} {$j < n} {incr j} {
        set b LSR$j
        eval $ns LDP-peer $$a $$b
}
set m [eval $$a get-module "MPLS"]
```

Za vsako vozlišče z vklopljenim protokolom MPLS je treba nastaviti agent LDP. Agent LDP omogoča izvajanje protokola LDP. Ker je vozlišč, ki imajo vklopljen protokol MPLS, lahko več, se agenti LDP nastavijo v zanki. Nastavijo se z uporabo ukaza "*LDP-peer $$a $$b*", s katerim določimo sosednja vozlišča v LDP-seji.

3. Ukazi za določitev načina razširjanja in dodeljevanja label

Modul MPLS podpira različne ukaze za določitev načina razširjanja in dodeljevanja label v vozliščih. Načine lahko določimo za vsa vozlišča MPLS ali za posamezna vozlišča (npr. MPLSnode).

Za vsa vozlišča MPLS so načini določeni z naslednjimi ukazi:

$ns enable-control-driven ali Classifier/Addr/MPLS set control_driven_ 1
$ns enable-data-driven ali Classifier/Addr/MPLS enable-data-driven
$ns enable-on-demand ali Classifier/Addr/MPLS enable-on-demand
$ns enable-ordered-control ali Classifier/Addr/MPLS enable-ordered-control

Z ukazom "*enable-control-driven*" določimo kontrolno voden način, kjer se labele razširjajo pred prihodom paketov v vozlišče MPLS. Ukaz "*enable-data-drive*" omogoča podatkovno voden način, ki omogoča razširjanje label s prihodom paketa v vozlišče MPLS. Ukaz "*enable-on-demand*" omogoča dodeljevanje label na zahtevo. Ukaz "*enable-ordered-control*" omogoča dodeljevanje in razširjanje label v urejenem načinu.

Za izbrana vozlišča MPLS so določeni načini z naslednjimi ukazi:

[$MPLSnode get-module "MPLS"] enable-control-driven
[$MPLSnode get-module "MPLS"] enable-data-driven
[$MPLSnode get-module "MPLS"] enable-on-demand
[$MPLSnode get-module "MPLS"] enable-ordered-control

4. Ukazi za sledenje dogajanja v vozlišču MPLS

Če želimo preveriti delovanje vozlišča MPLS, uporabimo ukaz "*trace-mpls*", s katerim dobimo podatke o dogajanju v vozlišču MPLS. Primer ukaza:

[$MPLSnode get-module "MPLS"] trace-mpls.

Primer izpisa je prikazan na sliki 26. Izpis je sestavljen iz vrstic, ki vsebujejo različna polja. Prvo polje pomeni simulacijski čas v sekundah in označuje čas proženja dogodka, ki predstavlja prihod paketa v vozlišče MPLS. Drugo polje predstavlja vozlišče MPLS, kamor je prispel paket. Sledi polje, ki predstavlja izvor in cilj paketa. Temu polju sledi polje, ki označuje, ali je paket opremljen z labelo (L) ali ne (U). Naslednje polje je vrednost vhodne labele. Sledi polje, ki predstavlja operacijo nad labelo, ki je lahko dodelitev (push), odvzem (pop) in zamenjava (swap). Sledita polji, ki predstavljata izhodni vmesnik in izhodno labelo. Zadnji dve polji predstavljata velikost glave MPLS in polje TTL v labeli.

```
0.1796  1: 0->8  U  -1  Push(ingress)     3  1  32  4
0.1912  3: 0->8  L   1  Swap              5  1  31  4
0.1948  5: 0->8  L   1  Pop(penultimate)  7  0  30  0
```

Slika 26: Izpis podatkov za vozlišče MPLS [12]

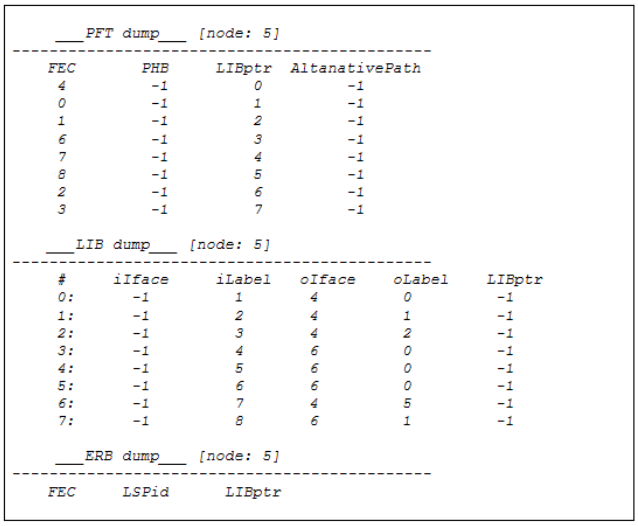
5. Prikaz informacij o tabelah MPLS

Vsako vozlišče MPLS vsebuje tabele, ki so opisane v poglavju 7.2.2.

Za prikaz vrednosti tabel na izbranem vozlišču MPLS uporabimo ukaze:

[$MPLSnode get-module "MPLS"] pft-dump
[$MPLSnode get-module "MPLS"] lib-dump
[$MPLSnode get-module "MPLS"] erb-dump

Ukaz "*pft-dump*" prikazuje vrednost tabele PFT. Z ukazom "*lib-dump*" prikažemo vrednost tabele LIB. Ukaz "*erb-dump*" prikazuje stanje tabele ERB. Primer izpisa vseh tabel za poljubno vozlišče je na sliki 27. Iz slike vidimo, da je tabela ERB prazna, saj ni vzpostavljene eksplicitne poti.



___PFT dump___ [node: 5]

FEC	PHB	LIBptr	AltanativePath
4	-1	0	-1
0	-1	1	-1
1	-1	2	-1
6	-1	3	-1
7	-1	4	-1
8	-1	5	-1
2	-1	6	-1
3	-1	7	-1

___LIB dump___ [node: 5]

#	iIface	iLabel	oIface	oLabel	LIBptr
0:	-1	1	4	0	-1
1:	-1	2	4	1	-1
2:	-1	3	4	2	-1
3:	-1	4	6	0	-1
4:	-1	5	6	0	-1
5:	-1	6	6	0	-1
6:	-1	7	4	5	-1
7:	-1	8	6	1	-1

___ERB dump___ [node: 5]

FEC	LSPid	LIBptr

Slika 27: Izpis tabel PFT, LIB in ERB

6. Ukazi za vzpostavitev in sprostitev eksplicitne poti

Za upravljanje poti v omrežju MPLS so potrebni ukazi za vzpostavitev in sprostitev eksplicitnih poti. Eksplicitno pot vzpostavimo z ukazom:

[$MPLSnode get-module "MPLS"] make-explicit-route fec ER LSPid rc.

Ukaz "*make-explicit-route*" uporabimo na vozlišču, kjer želimo začeti z vzpostavitvijo eksplicitne poti. Za celotno vzpostavitev eksplicitne poti je treba določiti parametre ukaza "*make-explicit-route*". Prvi parameter fec označuje ciljno (končno) vozlišče za eksplicitno pot. Parameter ER predstavlja listo vozlišč, med katerimi je treba vzpostaviti eksplicitno pot. S parametrom LSPid določimo identifikacijsko številko, s katero označujemo vzpostavljeno pot LSP. Zadnji parameter rc ima vedno privzeto vrednost -1.

Vsako vzpostavljeno eksplicitno pot lahko sprostimo z naslednjim ukazom:

[$MPLSnode get-module "MPLS"] ldp-trigger-by-release fec LSPid.

Ukaz "*ldp-trigger-by-release*" sprosti vzpostavljeno eksplicitno pot. Uporabi se nad vozliščem, ki predstavlja začetek eksplicitne poti. Za sprostitev eksplicitne poti je treba določiti parametre, kot sta fec in LSPid. S parametrom fec določimo končno vozlišče eksplicitne poti. Parameter LSPid predstavlja identifikacijsko številko eksplicitne poti, ki jo želimo sprostiti.

8 Metoda dela za izvajanje simulacije z NS-2

Glavni koraki za simulacijo z NS-2 [31]:

- **Korak 1:** Načrt simulacije
  Prvi korak simulacije omrežja je načrt simulacije. V tem koraku se določijo simulacijski model, namen simulacije in predpostavke, ki jih upoštevamo pri izvajanju simulacij. Korak 1 vključuje izbiro performančnih parametrov in predpostavke rezultatov merjenja.
- **Korak 2:** Nastavitev in izvajanje simulacije
  Ta korak implementira načrt, ki smo si ga zastavili v prvem koraku. Sestavljen je iz dveh faz:
  - Faza nastavitve omrežja: ustvarimo in nastavimo komponente omrežja (npr. vozlišča TCP, UDP), glede na načrt simulacije. Prav tako se nastavijo dogodki, ki se prožijo v določenem času. Primer takšnega dogodka je generiranje paketov.
  - Faza simulacije: prične simulacijo, ki je predhodno nastavljena. V tej fazi se dogodki izvajajo po kronološkem vrstnem redu. Faza se izvaja, dokler simulacijska ura ne poteče.
- **Korak 3:** Analiza po simulaciji
  Glavna naloga tega koraka sta preverjanje pravilnosti programa in ovrednotenje zmogljivosti simuliranega omrežja. Prva naloga se nanaša na odpravljanje napak v programu, druga pa na zbiranje in sestavljanje simulacijskih rezultatov.

8.1 Določitev gradnikov topologije omrežja

S programskim jezikom OTcl zgradimo topologijo omrežja in definiramo proženje dogodkov v omrežju. Izgradnjo vozlišč in povezav omogoča objekt simulator. Objekt simulatorja vsebuje funkcije, ki so potrebne za izgradnjo vozlišč, povezav. Tako lahko z instanco $ns ustvarimo nova vozlišča in povezave. Vozlišča, ki jih ustvarimo, so lahko tipa unicast (enouporabniško usmerjanje) ali tipa multicast (več uporabniško usmerjanje). Za ustvarjanje instance objekta se uporabi ukaz:

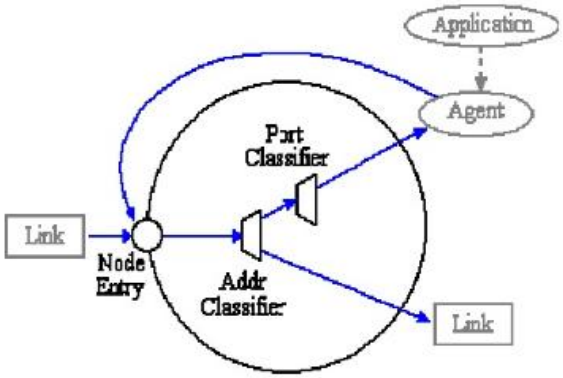
set ns [new Simulator].

Vozlišče tipa unicast ustvarimo z ukazom:

set n0 [$ns node].

Zgradba vozlišča, ki ga ustvarimo, je prikazana na sliki 28. Vozlišče je sestavljeno iz objektov:

- Node Entry: predstavlja stičišče povezave in vozlišča,
- Addr Classifier: razvršča pakete na naslednjo povezavo, če le-ta obstaja oziroma do razvrščevalnika vhodov,
- Port Classifier: posreduje pakete do agenta,
- Agent: predstavlja končno točko, ki ustvari ali uniči paket.

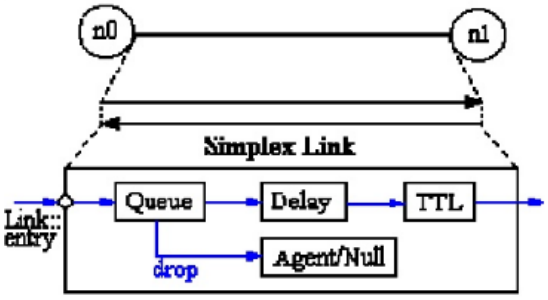


Slika 28: Zgradba vozlišča [17]

Če je vozlišče končna točka v omrežju, npr. ponor ali izvor, je vsakemu vozlišču treba dodati objekt agent. Tip objekta agent je odvisen od transportnega protokola, ki je potreben pri pošiljanju omrežnega prometa. Če aplikacija generira omrežni promet, ki za pošiljanje potrebuje transportni protokol UDP, je tudi agent tipa UDP.

Vozlišča med seboj povežemo s povezavo. Povezava je lahko enosmerna oziroma dvosmerna. Če nastavimo dvosmerno povezavo, sta vzpostavljeni dve enosmerni povezavi. Na sliki 29 je prikazana dvosmerna povezava med vozliščema n0 in n1. Enosmerna povezava je sestavljena iz objektov:

- Link entry: vhod v povezavo.
- Queue: vhod in izhod iz strežne vrste. Izpuščeni paketi se iz strežne vrste pošljejo do agenta z izhodom nič, kjer se sprostijo.
- Delay: za določanje zakasnitve.
- TTL: izračun parametra TTL za vsak paket posebej.

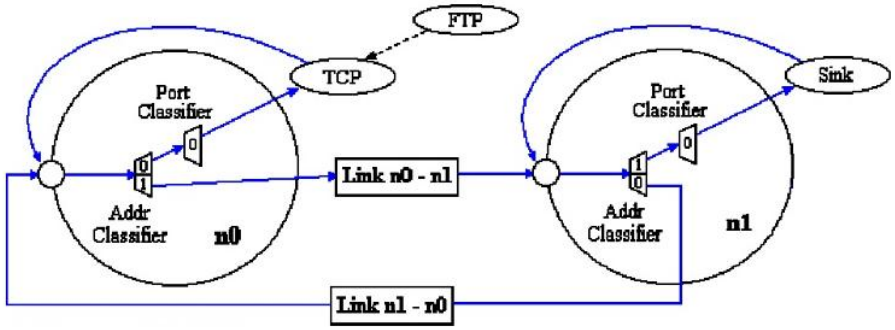


Slika 29: Zgradba povezave [17]

Povezavi med dvema vozliščema določimo pasovno širino, zakasnitev in tip čakalne vrste. Čakalna vrsta je lahko različnih tipov. Za simulacijo modelov bomo uporabili čakalno vrsto z odmetavanjem zadnjega konca (DropTail). To pomeni, da se bodo paketi pri zapolnjeni čakalni vrsti izpustili na koncu čakalne vrste. Dvosmerne povezavo med vozliščem n0 in n1 ustvarimo z ukazom:

$ns duplex-link $n0 $n1 <pasovna širina> <zakasnitev> <čakalna vrsta>.

Primer povezovanja dveh vozlišč v omrežnem simulatorju NS-2 je prikazano na sliki 30. Vozlišči n0 in n1 povežemo med seboj z dvosmerno povezavo. Vozlišče n0 je generator omrežnega prometa, vozlišče n1 pa ponor oziroma cilj omrežnega prometa.



Slika 30: Primer povezave dveh vozlišč [17]

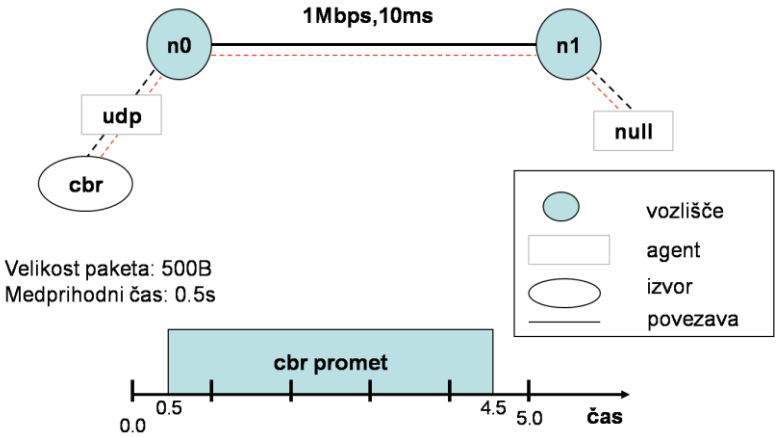
Ko imamo zgrajeno topologijo omrežja, lahko pošiljamo omrežni promet skozi omrežje. Začetek procesa pošiljanja omrežnega prometa nam omogoča razvrščevalnik dogodkov, ki se kreira ob instanci objekta simulator. Z razvrščevalnikom dogodkov ob točno določenem času prožimo določen dogodek. Primer ukaza za proženje dogodka ob točno določenem času je:

$ns at <cas> <dogodek>.

Čas, pri katerem sprožimo določen dogodek, je podan v sekundah. Dogodek je lahko pričetek ali prekinitev generiranja omrežnega prometa, zaključek simulacije itd.

8.2 Generiranje omrežnega prometa

Generiranje omrežnega prometa je mogoče na vozliščih, ki jih ustrezno nastavimo. Na vozliščih generiranje prometa nastavimo na dveh nivojih: na transportnem in aplikacijskem. Na aplikacijskem nivoju izberemo aplikacijo, s katero želimo tvoriti omrežni promet. Aplikacija je lahko CBR, FTP, telnet, itd. Glede na izbrano aplikacijo izberemo transportni protokol, ki ga zahteva aplikacija. Če želimo uporabiti aplikacijo FTP, moramo uporabiti protokol TCP, za aplikacijo CBR pa uporabimo protokol UDP. Transportni protokol določimo v objektu agent. Izbrano aplikacijo in agenta povežemo z ukazom "*attach-agent*", ki se uporabi pri nastavitvi agenta v vozlišču. Primer povezave aplikacije CBR in agenta UDP nad vozliščem n0 je na sliki 31.



Slika 31: Povezava med aplikacijo CBR in agentom protokola UDP

Primer ukazov za nastavitev vozlišča n0 (slika 31), ki generira omrežni promet:

set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp

Poleg ukazov, ki omogočajo generiranje prometa, je pri aplikaciji CBR treba določiti tudi velikost paketov in medprihodni čas paketov, kot na sliki 31. Za aplikacijo FTP, ki deluje preko protokola TCP, je mogoče nastaviti le velikost paketa in širino okna, ki določa, koliko paketov se bo preneslo po vzpostavljeni povezavi, preden bo sprejemnik poslal potrdilo. Medprihodnega časa za aplikacijo FTP ni mogoče določiti zaradi lastnosti protokola TCP, ki prenese pakete ob potrditvi.
Če želimo generirati omrežni promet, je treba določiti ponor omrežnega prometa in ga povezati z izvorom, kot je prikazano z rdečo povezavo na sliki 31. To se naredi z ukazi:

set null [new Agent/Null]
$ns attach-agent $n1 $null
$ns connect $udp $null

Izvajanje simulacije poteka v končnem času. V tem času se generira omrežni promet, kot cbr promet na sliki 31. Omrežni promet se generira po pretečenem času 0,5 sekunde. Pričetek generiranja omrežnega prometa za aplikacijo CBR se sproži z ukazom:

$ns at 0.5 "$cbr start".

Pred zaključkom simulacije je treba omrežni promet ustaviti. Na sliki 31 se omrežni promet ustavi po preteku časa 4,5 sekunde. To se stori z ukazom:

$ns at 4.5 "$cbr stop".

Izkušnje simulacij z omrežnim simulatorjem NS-2 so pokazale, da je generiranje prometa treba ustaviti, preden končamo simulacijo, sicer dobimo nepričakovan pojav na koncu izvajanja simulacije, ker ostanejo zahteve v sistemu nepostrežene. Posledica tega je napačno vrednotenje rezultatov.

8.3 Sledenje omrežnega prometa

Če želimo spremljati dogajanje v omrežju, je treba skozi simulacijo slediti toku omrežnega prometa s sledilno datoteko. Sledilna datoteka je tekstovna datoteka, v kateri so zapisi za vsak paket posebej. Zapis predstavlja podrobnosti o tem, kateri mejnik (npr. vozlišče in vrsta) je paket prečkal pri prenosu skozi omrežje. Sledilno datoteko generiramo z ukazom:

$ns trace-all $datoteka.

Sledilne datoteke se uporabijo pri procesu analize podatkov. Iz sledilne datoteke izberemo podatke, ki jih potrebujemo za analizo in vrednotenje omrežja. Podatke, ki nas zanimajo, dobimo s programom, ki je napisan s programskim jezikom AWK.

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

```
r : receive (at to_node)                    src_addr : node.port (3.0)
+ : enqueue (at queue)                      dst_addr : node.port (0.0)
- : dequeue (at queue)
d : drop    (at queue)
```

```
r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Slika 32: Primer tekstovnega zapisa sledilne datoteke [17]

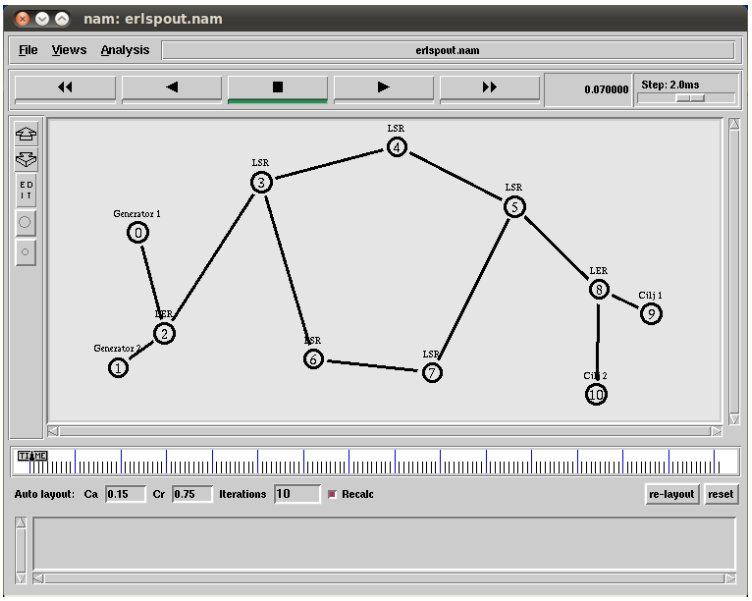
Format vsake sledi je sestavljen iz dvanajstih polj (slika 32). Vsako polje ima svoj pomen. Prvo polje je tip dogodka, ki ga je opravil paket. Tip dogodka se označuje z enim izmed štirih mogočih simbolov:

- r : sprejem paketa v vozlišču,
- + : vhod v vrsto,
- - : izhod iz vrste in
- d : paket je bil izpuščen in ni bil sprejet.

Drugo polje prikazuje čas, ob katerem se dogodek zgodi. Tretje in četrto polje predstavljata začetno in končno vozlišče na povezavi pri določenem dogodku. Peto polje pomeni tip paketa, ki je odvisen od vrste omrežnega prometa (npr. CBR, ali TCP). Šesto polje navaja velikost paketa. Sedmo polje je niz zastavic, ki se postavijo v izjemnih primerih. Osmo polje je identifikacijska številka toka prometa, ki jo določi uporabnik. Uporablja se za analizo, kjer z identifikacijsko številko lažje razločimo tok omrežnega prometa. Polji 9 in 10 označujeta izvorni in ciljni naslov. Polje enajst predstavlja sekvenčno številko omrežnega protokola. Polje dvanajst je zadnje polje v formatu sledi in označuje identifikacijsko številko paketa. Z identifikacijsko številko paketa določimo kraj in čas nahajanja paketa v omrežju.

8.4 Animacijsko orodje NAM

Z animacijskim orodjem NAM (ang. Network AniMation) si pomagamo pri preverjanju pravilnosti delovanja simulacijskega modela. Poleg tega lahko z njim spremljamo omrežni promet in celotno delovanje omrežja. Animacijsko orodje NAM je preprost uporabniški vmesnik, ki se uporablja za sledenje prometa in omogoča mnogo vizualnih značilnosti. Te značilnosti so barvni toki paketov, zakasnitev in izpuščanje paketov, označevanje vozlišč, barvanje povezav in prikaz delovanja vrste. Orodje NAM, ki je prikazano na sliki 33, se vključi z ukazom "*nam*".



Slika 33: Primer orodja NAM

Za prikaz delovanja simulacije potrebuje orodje NAM sledilno datoteko, ki jo tvorimo s spodnjim ukazom:

$ns namtrace-all $datoteka.

Izvajanje simulacije v orodju NAM poteka po zapisih, ki so podani v datoteki.

8.5 Analiza simulacije omrežnega prometa

Analizo simulacije izvedemo po generiranju sledilne datoteke, v kateri so zapisi vsakega paketa. Daljši kot je čas izvajanja simulacije, večji je obseg podatkov v sledilni datoteki. Zaradi tega je statistika podatkov bolj točna. Analiza se izvaja s programom v programskem jeziku AWK, ki je pomemben za pridobitev ustreznih informacij iz sledilne datoteke in pravilno vrednotenje rezultatov simulacije. AWK je programski jezik, ki se uporablja za procesiranje tekstovnih datotek. Z ukazi v programu AWK obdelujemo zapise v tekstovni datoteki. Vsak zapis je sestavljen iz polj, ki so med seboj ločena s presledki. S programom AWK izberemo polja, ki nas zanimajo, in nad temi polji izvedemo operacije za izračun rezultatov. V diplomski nalogi se osredotočimo na izračun rezultatov za parametre kakovosti storitev, kot so zakasnitev, sprememba zakasnitev, izguba paketov in prepustnost. Za vsakega izmed njih napišemo svoj program v programskem jeziku AWK.

S programom AWK dobimo potrebne podatke, ki jih zapišemo v tekstovno datoteko in jih prikažemo na grafu. Z grafičnim prikazom podatkov postane analiza rezultatov simulacij enostavna. Za grafični prikaz podatkov uporabljamo orodje Gnuplot. Prednost programa AWK je v preprosti zgradbi in funkcionalnosti. Analiza sledilnih datotek s programom AWK ima tudi nekaj slabosti, ki so:

- pri tvorjenju sledilne datoteke se uporabi velika količina resursov (npr. prostora v pomnilniku), kar dramatično poveča simulacijski čas,
- sledilne datoteke vsebujejo veliko informacij, ki jih je težko analizirati,
- potrebno je dobro poznavanje programskega jezika AWK.

8.6 Parametri kakovosti storitev

Aplikacije iz podatkovnega sveta, kot so prenos elektronske pošte, datotek, in brskanja po svetovnem spletu, so večinoma nezahtevne glede kakovosti pri prenosu [8]. Večini je dovolj že povezljivost. Z napredkom tehnologij so na podatkovnih omrežjih nastale aplikacije, ki jim zgolj povezljivost ne zadošča več. Takšne aplikacije so na primer: prenos govora, prenos videa in igranje interaktivnih iger preko interneta. Aplikacije se izvajajo v realnem času, zato od omrežja pričakujejo izpolnjevanje določenih zahtev glede parametrov kakovosti storitev. Parametre kakovosti storitev bomo skozi simulacijo spremljali za vsak poslan paket posebej.

Parametri kakovosti storitev:

Zakasnitev (ang. delay) paketa predstavlja razliko med začetnim časom prenosa paketa in časom, ko je paket prispel na cilj (končni čas). Zakasnitev ni samo čas, ki se porabi za prenos paketa skozi omrežje, ampak je tudi čas paketa v strežni vrsti in procesorski čas obdelave paketa na vozlišču (strežniku). Merimo jo v milisekundah. Zakasnitev paketa izračunamo po naslednji formuli:

zakasnitev paketa = čas prihoda paketa na cilj – čas začetka prenosa. (1)

Spremenljivost zakasnitve (ang. jitter) predstavlja spremenljivost zakasnitve med dvema paketoma. Merimo jo v milisekundah. Oceno za jitter [33] izračunamo po naslednji enačbi:

$J(i) = J(i-1) + \left(\frac{|D(i-1,i)| - J(i-1)}{16}\right)$, kjer je: (2)

J(i) – vrednost jittra za trenutni paket i,
J(i-1) – vrednost jittra za prejšnji paket paketa i,
D(i-1,i) – razlika v zakasnitvi med trenutno poslanim paketom i in predhodno poslanim paketom paketa i.

Razlika v zakasnitvi D(i,j) oziroma sprememba zakasnitve je razlika med časom prenosa paketa i in časom prenosa paketa j. Računa se po naslednji enačbi:

D(i,j) = (Rj-Sj) - (Ri-Si), kjer je: (3)

Rj – čas prihoda paketa j na cilj,
Sj – začetni čas oddaje paketa j,
Ri – čas prihoda paketa i na cilj,
Sj – začetni čas oddaje paketa i.

Jitter (enačba 2) se računa iterativno za vsak sprejeti paket na koncu vozlišča. Vrednost jittra se izračuna na vsakem koraku in je odvisna od prejšnje vrednosti jittra ter razlike v zakasnitvi (enačba 3) med dvema oddanima in prejetima paketoma. Razliko med dvema oddanima in prejetima paketoma ter vrednostjo jittra prejšnjega paketa delimo z vrednostjo šestnajst, da zmanjšamo vpliv šuma.

Prepustnost (ang. Throughput) predstavlja, kolikšno število bitov se bo preneslo v določenem časovnem intervalu. Prepustnost računamo ob vsakem prejetem paketu. Časovni interval bo v našem primeru predstavljal razliko med začetkom pošiljanja paketa in prihodom vsakega paketa na cilj. Prepustnost merimo v megabitih na sekundo. Izračuna se po enačbi 4:

$$\text{prepustnost} = \left(\frac{poslani\ paketi\ v\ bitih}{simulacijski\ čas} \right) . \quad (4)$$

Delež izgubljenih paketov (ang. Packet loss) opisuje, koliko paketov je bilo izgubljenih med prenosom od izvora (vhoda) do cilja (izhoda). Delež izgubljenih paketov se meri v procentih. Parameter je izrednega pomena za realno časovni promet, ki se prenaša preko protokola UDP, ker ne zagotavlja dostave paketa. Zaradi tega razloga izguba vsakega paketa vpliva na delovanje realno časovne aplikacije. Izračuna se po enačbi 5:

$$\text{delež izgubljenih paketov} = \left(\frac{vsi\ izgubljeni\ paketi}{vsi\ poslani\ paketi} \right) * 100 . \quad (5)$$

9 Simulacije omrežja in analiza rezultatov

Za izvajanje simulacij omrežja IP s protokolom MPLS in brez njega smo uporabili omrežni simulator NS-2. Omrežni simulator NS-2 poganjamo na operacijskem sistemu Ubuntu Linux, ki je najbolj priljubljena distribucija Linuxa. S simulatorjem NS-2 smo preverili funkcionalnosti in vpliv protokola MPLS na omrežje IP z vidika parametrov kakovosti storitev, kot so zakasnitev prenosa paketov, sprememba zakasnitve, prepustnost in izguba paketov. Za ta namen smo preko eksperimentov zgradili različne simulacijske modele. Začeli smo s preprostim simulacijskim modelom z enostavno topologijo omrežja, s katerim smo preverili hitrost prenosa paketov s protokolom MPLS in brez njega. Model smo razširili v bolj kompleksnega z namenom, da analiziramo vpliv prometnega inženiringa, ki ga omogoča protokol MPLS z vzpostavitvijo eksplicitne poti skozi omrežje MPLS.

Predpostavke pri izvajanju simulacij

Realno omrežje MPLS je sestavljeno iz usmerjevalnikov, ki so med seboj povezani s kabli (npr. bakreni ali optični). Namesto izgradnje realnega omrežja smo za izvajanje eksperimentov definirali simulacijski model. Ta je lahko sestavljen iz vozlišč IP ali vozlišč MPLS, ki imajo omogočen protokol MPLS. Vozlišča so med seboj povezana s povezavami. Za izvajanje simulacije nad simulacijskim modelom predpostavimo:

1. Na vozlišču se lahko naenkrat obdeluje samo en paket oziroma ena zahteva. Pri obdelavi paketa na vozlišču ne prihaja do napak. Čas zadrževanja zahteve v sistemu je odvisen od časa čakanja v čakalni vrsti in od časa strežbe v vozlišču. Čas strežbe na vozlišču je konstanten.
2. Paketi se v strežnih vrstah razvrščajo statično, po principu FIFO (ang. First in First Out), kar pomeni, da izhajajo iz vrste v enakem vrstnem redu, kot v njo prihajajo.
3. Pri prenosu paketov med vozlišči ne prihaja do napak. Za vse pakete, ki se prenašajo v omrežju simulacijskega modela, je vnaprej določena pot, po kateri potujejo. Skozi simulacijski čas se le-ta ne spreminja.
4. Dolžina strežne vrste je končna. Privzeto največje število paketov, ki je lahko v strežni vrsti, je nastavljeno na vrednost 50. Pri zapolnjeni strežni vrsti se paketi izgubijo.
5. Po zapolnitvi strežne vrste se zahteve začnejo odmetavati po principu zadnjega konca (ang. DropTail). To pomeni, da se izpustijo vsi paketi, ki pridejo v zapolnjeno strežno vrsto.
6. V simulaciji smo se omejili na spremljanje samo ene vrste omrežnega prometa. Na eno vrsto omrežnega prometa smo se osredotočili zaradi manj kompleksne analize rezultatov.
7. Vrednost intenzivnosti pošiljanja omrežnega prometa se določi na začetku izvajanja simulacije.
8. Izvajanje simulacije poteka v končno določenem času. Privzeli smo, da se simulacija konča, ko preteče 20 sekund. V tem času se generira omrežni promet paketov, ki se uporablja za potrebe simulacije.

Predpostavke rezultatov

Osnovna predpostavka je, da se bodo v omrežju IP s protokolom MPLS paketi prenašali hitreje kot v omrežju IP brez protokola MPLS. V omrežju IP bodo vozlišča, ki imajo vklopljen protokol MPLS, prenašala pakete hitreje, z manjšo zakasnitvijo in z večjo prepustnostjo paketov.

Glavna aplikacija protokola MPLS je prometni inženiring. Omrežni simulator NS-2 omogoča prometni inženiring z vzpostavitvijo dodatne eksplicitne poti. Z uporabo prometnega inženiringa v omrežju MPLS predpostavljamo, da bodo parametri kakovosti storitev boljši kot brez uporabe prometnega inženiringa v omrežju IP. Tako bomo dobili manjšo zakasnitev, spremenljivost zakasnitve in izgubo paketov. Domnevamo, da bo z uporabo prometnega inženiringa prepustnost paketov konstanta.

Omejitve pri izvajanju simulacij

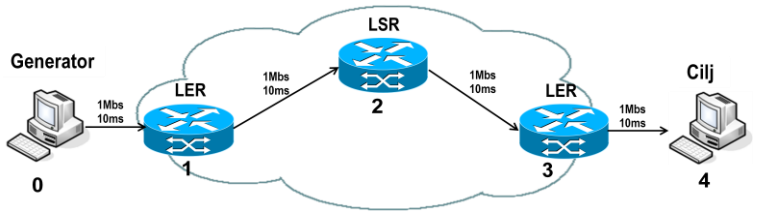
Omejitve simulacijskega modela glede na omrežje MPLS realnega sistema so:

1. Za vsa vozlišča v simulacijskem modelu velja enak čas procesiranja ne glede na intenzivnost prihajanja paketov. Vozlišča ne morejo dodeljevati procesnih virov kot usmerjevalniki v realnih omrežjih. Na usmerjevalniku se procesorski čas spreminja v odvisnosti od intenzivnosti prihajanja paketov.
2. V realnem omrežju MPLS je čas procesiranja IP-paketa, opremljenega z labelo MPLS, hitrejši kot na vozliščih omrežnega simulatorja, ker se uporablja strojna oprema, ki dejansko loči, ali se IP-paket posreduje na osnovi labele ali ne.
3. V realnem omrežju je lahko poljubno število uporabnikov, ki generirajo promet s pošiljanjem podatkov v omrežje v obliki paketov. V simulacijskem modelu smo se omejili na največ dva generatorja prometa.
4. V simulacijskem modelu ne moremo simulirati govornega prometa z uporabo aplikacije VoIP, kot je na primer program Skype, saj smo pri tem omejeni s strani omrežnega simulatorja NS-2. Omrežni simulator ne omogoča aplikacije VoIP, zato s simulacijo ne moremo preveriti, kako bi se naš simulacijski model odzival pri uporabi aplikacije VoIP in ostalega naključno generiranega omrežnega prometa. Omrežni simulator NS-2 tudi ne ponuja dobrega približka delovanju aplikacije VoIP. Če bi želeli izvajati simulacijo nad simulacijskim modelom z uporabo aplikacije VoIP, bi morali izbrati drug omrežni simulator, kot je na primer OPNET. Simulacijo s programom Skype bi lahko izvajali na realnem omrežju.
5. Realna omrežja IP so že nekaj časa nadgrajena s protokolom MPLS. Zaradi tega je mogoče v realnem omrežju MPLS izvajati različne funkcionalnosti, ki jih omogoča protokol MPLS. V nasprotju z realnimi omrežji pa omrežni simulator NS-2 ne omogoča vseh funkcionalnosti protokola MPLS, temveč samo izvajanje prometnega inženiringa z vzpostavitvijo eksplicitne poti. Kakovost storitev, navidezna zasebna omrežja in preusmerjanje prometa ob padcu povezave v omrežnem simulatorju niso podprti.

Pri podanih predpostavkah in omejitvah sistemov smo najprej preverili vpliv protokola MPLS na delovanje omrežja IP, ki ga zgradimo z omrežnim simulatorjem NS-2. V ta namen smo izvedli simulacijo protokola MPLS.

9.1 Simulacija protokola MPLS

Eksperiment izvedemo z namenom, da s simulacijo omrežja IP ugotovimo, kako protokol MPLS vpliva na omrežje IP. Zanima nas, ali se paketi skozi omrežje IP posredujejo hitreje ali ne. Predpostavljamo, da se bodo paketi skozi omrežje IP s protokolom MPLS prenašali hitreje kot v omrežju IP brez protokola MPLS. To bomo najlažje ugotovili tako, da bomo izmerili povprečno zakasnitev paketa od izvora do ponora. Hitrost prenosa paketov je namreč odvisna od zakasnitev paketov skozi omrežje. Večja kot je zakasnitev, manjša je hitrost prenosa in obratno. Hiter prenos paketov skozi omrežje pomeni večjo prepustnost poslanih paketov v določenem času.



Slika 34: Model preprostega omrežja IP

Eksperiment smo izvedli z izgradnjo preprostega modela omrežja IP s slike 34, ki je sestavljeno iz izvora (generator) in ponora (cilj) omrežnega prometa ter treh vozlišč IP, med katerimi sta dve na robu omrežja, eno pa vmes med njima. Vse povezave v omrežju imajo zakasnitev 10 ms in pasovno širino 1 Mbit/s. Model omrežja IP s protokolom MPLS je enak kot na sliki 34. Razlika je samo v tem, da je na vozliščih ena, dva in tri omogočen protokol MPLS. Skozi model omrežja s protokolom MPLS in brez njega pošiljamo konstanten tok paketov velikosti 1000B. Pakete pošiljamo na vsake pol sekunde.

S simulacijo omrežja IP s protokolom MPLS in brez njega dobimo enake rezultate. Povprečna zakasnitev paketa, ki potuje skozi omrežje IP od izvora do ponora, je v obeh primerih 72 ms. Po rezultatih sklepam, da s simulacijo omrežja IP s protokolom MPLS in brez njega ne dosežemo hitrejšega prenosa paketov, ker je zakasnitev enaka tako za omrežje s protokolom MPLS kot za omrežje brez njega. Za vozlišča MPLS v omrežju MPLS velja enak strežni čas kot za vozlišča IP v omrežju IP. Vozlišča izvajajo osnovne funkcionalnosti protokola MPLS, ne pa dejanske funkcije dodajanja labele paketu, s katero vplivamo na hitrejši prenos paketov skozi omrežje. Če bi želeli hitrejši prenos paketov v omrežju MPLS, bi moral omrežni simulator omogočati parametre, s katerimi bi nastavili čas strežbe za vozlišča IP in vozlišča MPLS.

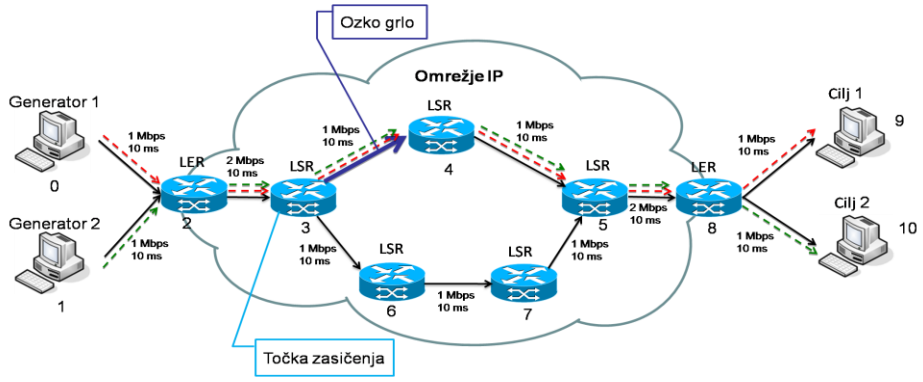
Kot smo ugotovili, na primeru enostavnega omrežja IP z uporabo MPLS in brez njega ne pridemo do oprijemljivih rezultatov, zato smo se v nadaljevanju osredotočili na uporabo protokola MPLS v povezavi s prometnim inženiringom.

9.2 Simulacija prometnega inženiringa

Vsi nadaljnji eksperimenti bodo temeljili na simulaciji prometnega inženiringa, ki s pomočjo protokola MPLS omogoča vzpostavitev eksplicitne poti skozi omrežje MPLS. V omrežju IP brez protokola MPLS prometni inženiring na osnovi vzpostavitve eksplicitne poti ni mogoč. Za potrebe izvajanja vseh nadaljnjih eksperimentov zgradimo dva simulacijska modela:

- model IP (slika 35),
- model MPLS (slika 36).

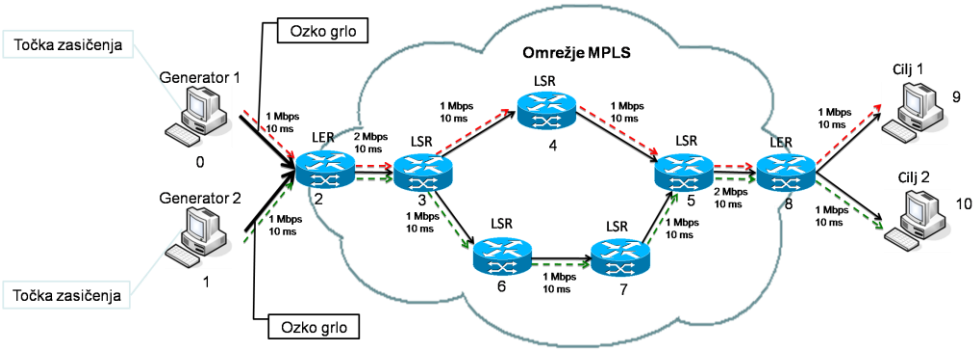
Značilnosti obeh simulacijskih modelov je enaka topologija omrežja z enakimi karakteristikami povezav. Simulacijska modela sta sestavljena iz dveh generatorjev omrežnega prometa, sedmih vozlišč (usmerjevalnikov) in dveh ciljev, ki predstavljata ponor omrežnega prometa. Vozlišča v omrežju so razporejena tako, da sta vozlišči 2 in 8 na robu omrežja, vozlišča 3, 4, 5, 6 ter 7 pa v hrbtenici omrežja. Tako predstavljajo vsa vozlišča med vozliščem 2 in 8 hrbtenično omrežje simulacijskega modela. Vse povezave v omrežju imajo zakasnitev 10 ms in pasovno širino 1 Mbit/s. Pasovno širino 2 Mbit/s imata povezavi med vozliščema 2 in 3 ter 7 in 8. Povezavi potrebujeta višjo pasovno širino od ostalih povezav, ker se na tem mestu promet združuje. Razlika med modelom IP in modelom MPLS je v vrsti uporabljenih vozlišč za izgradnjo omrežja. V modelu IP se uporabljajo običajna vozlišča IP, v modelu MPLS pa vozlišča s protokolom MPLS. Vozlišča MPLS omogočajo vzpostavitev eksplicitne poti skozi omrežje MPLS.
Kot je prikazano na sliki 35, se bo v modelu IP celotni omrežni promet iz generatorja 1 (rdeča črtkana puščica) in generatorja 2 (zelena črtkana puščica) pošiljal po najkrajši poti, ki poteka med vozlišči 2, 3, 4, 5 in 8. V modelu IP predpostavljamo, da bo prihajalo do točke zasičenja na vozlišču 3. Razlog za zasičenje bo v zmanjšanju pasovne širine iz 2 Mbit/s (povezava med vozliščem 2 in 3) na 1 Mbit/s (povezava med vozliščem 3 in 4). Povezava med vozliščem 3 in 4 (modra puščica na sliki 35) bo tako predstavljala ozko grlo hrbteničnega omrežja. Zaradi tega se bodo paketi pri veliki intenzivnosti pošiljanja nabirali v strežno vrsto vozlišča 3 (točka zasičenja), dokler ne bo prišlo do zapolnitve strežne vrste in s tem do izgube paketov. Izguba paketov je tako omejena z dolžino strežne vrste.



Slika 35: Model IP

V modelu MPLS na sliki 36 se bo omrežni promet iz generatorja 1 (rdeča črtkana puščica) pošiljal po najkrajši poti, omrežni promet iz generatorja 2 (zelena črtkana puščica) pa po eksplicitni poti. Eksplicitno pot med vozlišči 2, 3, 6, 7, 5 in 8 predhodno nastavimo z ukazi iz poglavja 7.2.4. Pri eksplicitni poti ni nujno, da je optimalna, saj lahko vsebuje več vozlišč in povezav med vozlišči, kar lahko poveča zakasnitev paketov na poti. Zaradi vzpostavitve dodatne poti v modelu MPLS predpostavimo, da bomo odpravili problem ozkega grla v hrbteničnem omrežju modela MPLS. Promet se bo iz povezave med vozliščema 2 in 3 s

pasovno širino 2 Mbit/s pošiljal na dve ločeni povezavi s pasovno širino 1 Mbit/s. Tako na vozlišču 3 ne bo točke zasičenja, pri kateri bi prišlo do zapolnitve strežne vrste in izgube paketov. Točka zasičenja se lahko pojavi na generatorju 1 in generatorju 2, saj je povezava med generatorjem in vozliščem 2 samo 1 Mbit/s in predstavlja ozko grlo. Generatorja lahko pošiljata pakete v omrežje z največjo hitrostjo 1 Mbit/s. Če je hitrost večja, bo prišlo do točke zasičenja že na generatorju in paketi se ne bodo posredovali v hrbtenično omrežje MPLS.

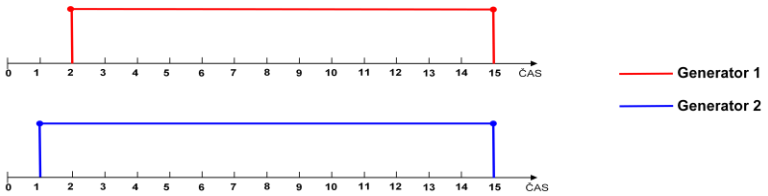


Slika 36: Model MPLS

Izvajanje simulacije bo potekalo v dvajsetsekundnem časovnem intervalu. V tem času se bo generiral omrežni promet iz generatorja 1 in generatorja 2. Omrežni promet iz obeh generatorjev prometa bomo generirali v različnih časovnih trenutkih enako za model IP kot za model MPLS. Promet iz generatorja 2 bomo generirali v simulacijskem času ene sekunde, kot prikazuje modra črta na sliki 37. Promet iz generatorja 1 bomo generirali v času dveh sekund, kot prikazuje rdeča črta na sliki 37. Generiranje prometa bomo ustavili po preteku simulacijskega časa petnajstih sekund.

Z generatorjem 1 bomo v eksperimentih generirali različne vrste prometa. Generirali bomo omrežni promet z deterministično in poissonovo verjetnostno porazdelitvijo preko protokola UDP ter promet FTP preko protokola TCP. Omrežni promet na generatorju 2 se bo v vseh eksperimentih generiral z deterministično porazdelitvijo. To pomeni, da se bodo paketi pošiljali z enakimi medprihodnimi časi.

Skozi simulacijo bomo opazovali, kako se parametri kakovosti storitev omrežnega prometa paketov generatorja 2 spreminjajo glede na začetek generiranja omrežnega prometa na generatorju 1 v modelu IP in modelu MPLS. Pri analizi vseh rezultatov simulacij prometnega inženiringa se bomo osredotočili na omrežni promet iz generatorja 2, saj je zaradi deterministične porazdelitve enostaven za prikaz in analizo.



Slika 37: Generiranje omrežnega prometa za generator 1 in generator 2

Namen vseh eksperimentov prometnega inženiringa je preveriti skozi simulacijo, kako uporaba eksplicitne poti v omrežju MPLS vpliva na parametre kakovosti storitev pri različni vrsti in intenzivnosti omrežnega prometa, pri čemer bomo uporabili modela IP in MPLS.

9.2.1 Eksperiment 1

Pri eksperimentu 1 se generira omrežni promet z deterministično porazdelitvijo na obeh generatorjih modelov IP in MPLS. Paketi omrežnega prometa se pošiljajo v konstantnih časovnih intervalih z uporabo aplikacije CBR, ki deluje preko protokola UDP. Velikost generiranega paketa je 512B. Za generator 1 in generator 2 so nastavljeni parametri prikazani v tabeli 3 in v času izvajanja eksperimenta ostajajo nespremenjeni. Spreminja se samo intenzivnost za generator 2, generator 1 ima enako intenzivnost za celotni eksperiment 1.

	Generator 1	Generator 2
Velikost paketa	512B	512B
Generiranje paketov	deterministično	deterministično
Aplikacija	CBR	CBR
Transportni protokol	UDP	UDP

Tabela 3: Parametri za eksperiment 1

$$\text{Največja prepustnost omrežja (paketi/s)} = \frac{hitrost\ povezave\ (Mbit/s)}{velikost\ paketa\ (\check{s}tevilo\ bitov)} \quad (6)$$

Za model IP eksperimenta določimo teoretično zgornjo mejo skupne intenzivnosti pošiljanja paketov iz generatorja 1 in generatorja 2, pri kateri ne bo prihajalo do točke zasičenja na vozlišču 3 (slika 35) in posledične izgube paketov. Teoretično zgornjo mejo oziroma največjo prepustnost omrežja izračunamo na podlagi velikosti paketa in hitrosti povezave med vozliščema 3 in 4 (enačba 6), ki je ozko grlo hrbteničnega omrežja modela IP (slika 35). Ker je velikost vsakega poslanega paketa 512B in pasovna širina povezave 1 Mbit/s, je teoretična zgornja meja približno 245 paketov/s. Če ne želimo točke zasičenja v hrbteničnem omrežju modela IP, mora biti celotna intenzivnost pošiljanja iz generatorja 1 in generatorja 2 manjša od intenzivnosti 245 paketov/s, sicer bo prišlo v modelu IP do zapolnitve strežne vrste na vozlišču 3 (slika 35), ki predstavlja točko zasičenja.

Teoretična zgornja meja skupne intenzivnosti pošiljanja paketov iz generatorja 1 in generatorja 2 je za model MPLS dvakrat večja kot za model IP, saj se promet v hrbteničnem omrežju pošilja po dveh ločenih poteh s pasovno širino 1 Mbit/s (slika 36). Tako bo teoretična zgornja meja skupne intenzivnosti v modelu MPLS približno 490 paketov/s.

V modelih IP in MPLS mora za generator 1 in generator 2 veljati, da pošiljata pakete z največjo intenzivnostjo 245 paketov/s. Največjo intenzivnost pošiljanja paketov določa povezava med generatorjem in vozliščem 2 (slika 36). Ker je velikost vsakega generiranega paketa 512B in pasovna širina povezave 1 Mbit/s, je lahko največja intenzivnost generatorja 245 paketov/s. Če je intenzivnost pošiljanja na generatorju večja od 245 paketov/s, bo prišlo do točke zasičenja že na generatorju (slika 36) ter s tem do izgube paketov, preden bodo prispeli v hrbtenično omrežje modelov IP in MPLS. Zaradi tega teoretično v hrbteničnem omrežju modela MPLS ne more priti do zasičenja in izgube paketov.

Generator 1 je pošiljal pakete s fiksno intenzivnostjo 100 paketov/s, za generator 2 pa smo določili različne intenzivnosti pošiljanja paketov (10, 50, 100, 160…). Pri različnih intenzivnostih generatorja 2 in fiksni intenzivnosti generatorja 1 smo izmerili povprečne zakasnitve pošiljanja paketov v modelih IP in MPLS (tabela 4). Poleg povprečnih zakasnitev smo izmerili tudi povprečne vrednosti jittra (tabela 5).

Generator 2 (paketi/s)	IP (ms)	MPLS (ms)
10	84,28	96,47
50	84,28	96,47
100	84,28	96,47
160	**241,02**	**94,76**
200	254,65	95,80
250	385,98	237,71

Tabela 4: Povprečna zakasnitev za eksperiment 1

Na podlagi tabele 4 in izmerjene povprečne zakasnitve smo določili intenzivnost za generator 2, pri kateri smo simulirali parametre kakovosti storitev za modela IP in MPLS. Iz tabele 4 vidimo, da se povprečna zakasnitev do intenzivnosti 160 paketov/s ne spreminja za modela IP in MPLS. Povprečna zakasnitev je za majhne intenzivnosti (npr. 10, 50, 100 paketov/s) manjša v modelu IP kot v modelu MPLS, zato ničesar ne pridobimo, če pošiljamo promet po eksplicitni poti v modelu MPLS. Na račun eksplicitne poti se bo v modelu MPLS pojavila celo večja zakasnitev. Do opazne razlike pridemo pri intenzivnosti 160 paketov/s, kjer se v modelu IP povprečna zakasnitev drastično poveča, v modelu MPLS pa celo za malenkost zmanjša. Zmanjšanje zakasnitve v modelu MPLS je posledica spremembe medprihodnega časa paketov v strežno vrsto na vozlišču 2 (slika 36), kjer se združuje omrežni promet iz generatorja 1 in generatorja 2.

Pri intenzivnosti 160 paketov/s na generatorju 2 se povprečna zakasnitev v modelu IP poveča zaradi točke zasičenja na vozlišču 3 (slika 35). Pri točki zasičenja se strežna vrsta zapolni in paketi se zavržejo. Točka zasičenja se pojavi, ker skupna intenzivnost 260 paketov/s iz generatorja 1 in generatorja 2 preseže teoretično zgornjo mejo skupne intenzivnosti 245 paketov/s. Za vsako nadaljnje povečanje intenzivnosti na generatorju 2 bo sledilo povečanje povprečne zakasnitve v modelu IP, saj se skupna intenzivnost iz generatorja 1 in generatorja 2 samo povečuje ter ne ustreza teoretični zgornji meji intenzivnosti 245 paketov/s.

V modelu MPLS se povprečna zakasnitev do intenzivnosti 250 paketov/s bistveno ne spreminja. Povprečna zakasnitev se poveča pri intenzivnosti 250 paketov/s, kar je posledica točke zasičenja na generatorju 2 (slika 36). Do točke zasičenja pride, ker je intenzivnost na generatorju 2 (250 paketov/s) večja od največje intenzivnosti generatorja (245 paketov/s).

Generator 2 (paketi/s)	IP (ms)	MPLS (ms)
10	0,029	0,001
50	0,005	0,008
100	0,002	0,001
160	1,978	0,346
200	1,548	0,411
250	1,314	0,122

Tabela 5: Povprečen jitter za eksperiment 1

Povprečna vrednost jittra (tabela 5) je v modelu MPLS vedno manjša kot v modelu IP, ne glede na intenzivnost pošiljanja omrežnega prometa na generatorju 2. Vzrok za manjši jitter v modelu MPLS je v tem, da se pošilja omrežni promet iz generatorja 2 po eksplicitni poti ločeno od ostalega omrežnega prometa, medtem ko se v modelu IP pošilja z ostalim prometom po isti (najkrajši) poti. Zaradi pošiljanja po ločenih poteh se v strežni vrsti na vozlišču 3 (slika 36) nabira manj paketov, posledica tega pa so manjše povprečne vrednosti jittra. Zanimiv pojav je pri 250 paketih/s, kjer vrednosti jittra padejo zaradi točke zasičenja na generatorju. Točka zasičenja na generatorju povzroči manjše število prenesenih paketov v hrbtenično omrežje modela IP in MPLS.

V nadaljevanju nas je zanimalo, kako generator 2 z intenzivnostjo 160 paketov/s in generator 1 z intenzivnostjo 100 paketov/s vplivata na simulacijo parametrov kakovosti storitev v modelih IP in MPLS. To je prikazano v scenariju 1. V scenariju 2 smo preverili za primer, ko generator 2 generira omrežni promet z intenzivnostjo 10 paketov/s.

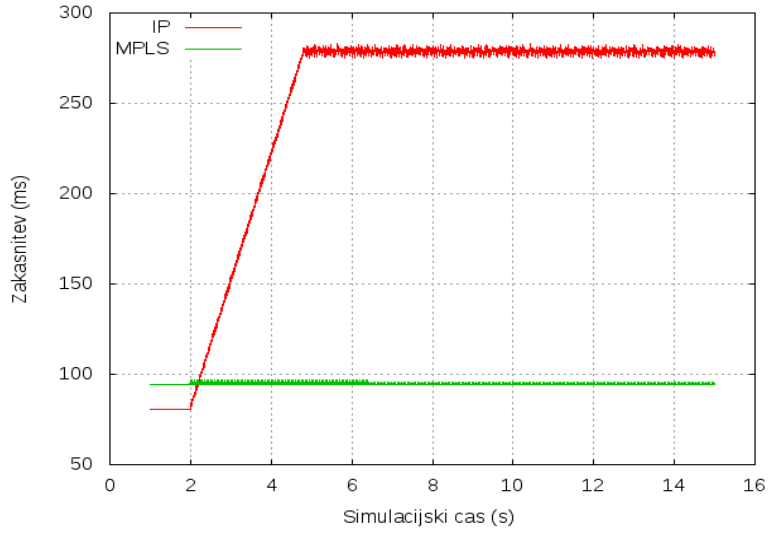
Scenarij 1

V scenariju 1 smo generirali omrežni promet s parametri iz tabele 3 in z naslednjimi intenzivnostmi za generator 1 in generator 2:

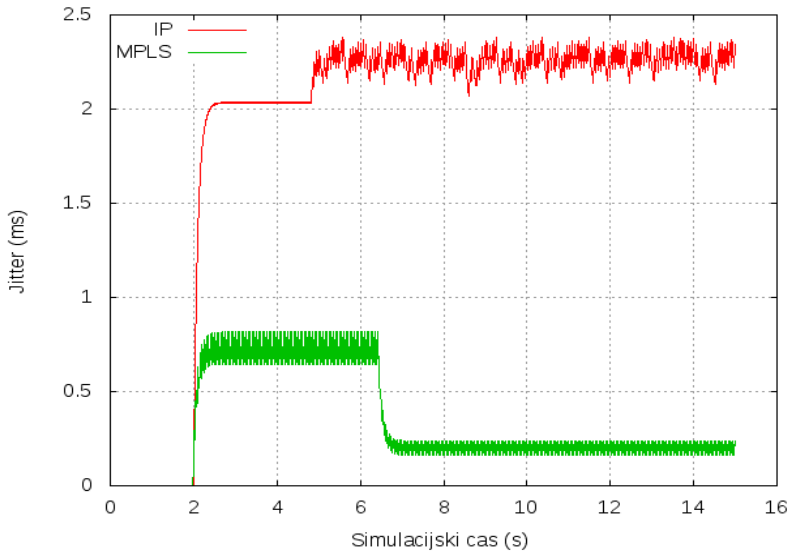
- generator 1: 100 paketov/s,
- generator 2: 160 paketov/s.

Za scenarij 1 pri eksperimentu 1 smo skozi čas simulacije opazovali parametre kakovosti storitev, ki so prikazani na sliki 38.

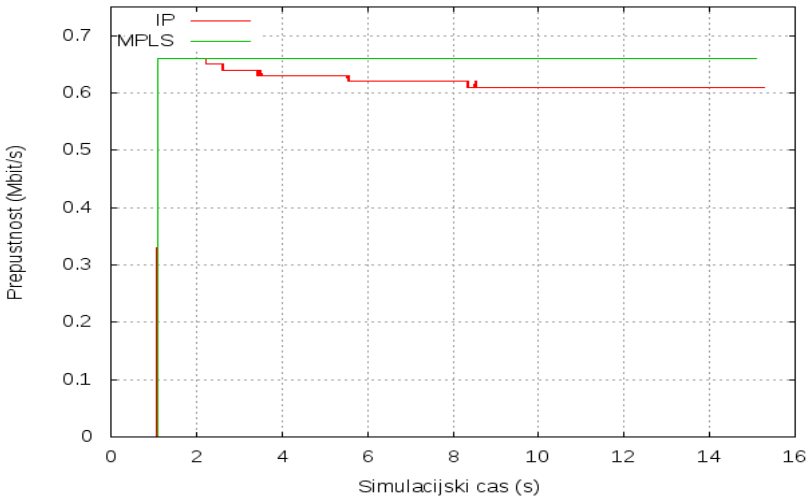
a)



b)



c)



Slika 38: Prikaz parametrov kakovosti storitev za omrežni promet iz generatorja 2 skozi čas simulacije pri scenariju 1 eksperimenta 1 za a) zakasnitev, b) jitter in c) prepustnost

Iz grafa a) na sliki 38 vidimo, da se do časa dveh sekund paketi v modelu IP pošiljajo z manjšo zakasnitvijo kot paketi v modelu MPLS. V času dveh sekund se začnejo pošiljati paketi omrežnega prometa iz generatorja 1. Promet iz generatorja 1 vpliva na spremembo zakasnitve paketov iz generatorja 2. Zakasnitev v modelu IP linearno raste od časa dveh do časa petih sekund. To pomeni kopičenje paketov v strežni vrsti vozlišča, dokler se le-ta ne zapolni. Ko se strežna vrsta zapolni, pridemo do točke zasičenja, kjer se začnejo paketi izgubljati. Zakasnitev preneha linearno rasti in začne variirati. Točka zasičenja se pojavi po času petih sekund na vozlišču 3 modela IP (slika 35).

$$\text{Medprihodni čas paketa (s)} = \frac{1}{intenzivnost\ pošiljanja\ (paketi/s)} \qquad (7)$$

V modelu MPLS je porast zakasnitve manjši, ker se ne pojavi točka zasičenja. Zakasnitev variira v majhnih mejah skozi celotni čas simulacije. Zanimiv pojav, ki se zgodi med časom šestih in osmih sekund, je komaj opazno zmanjšanje zakasnitve. Čeprav je sprememba zakasnitve majhna, je njen vpliv dobro viden na vrednosti jittra iz grafa b) na sliki 38. To pripišemo dejstvu, da se med časom šestih in osmih sekund spremeni medprihodni čas paketov (generator 2) v strežno vrsto na vozlišču 2 (slika 36). Medprihodni čas je z vsakim prihodom paketa v strežno vrsto krajši za neko majhno vrednost. V našem primeru je zaradi pošiljanja paketov iz generatorja 2 z intenzivnostjo 160 paketov/s medprihodni čas (enačba 7) vsakega poslanega paketa 0,00625 sekund. Ker je medprihodni čas izredno majhen in podan na pet decimalk natančno, se bo njegov vpliv v strežni vrsti poznal šele po pretečenem času šestih sekund. Tako se bo ta pojav ponavljal skozi čas simulacije na približno vsakih 6 sekund. Na krajšanje medprihodnega časa vplivata predvsem zadnji dve decimalki. Če izberemo medprihodni čas 0,006 sekund, do vpliva krajšanja medprihodnega časa ne bo prišlo, zato se ne pojavi sunkovit padec v vrednosti jittra.

Vpliv krajšanja medprihodnega časa pri intenzivnosti 160 paketov/s smo preverili s simulacijo. Pri tem smo spremljali, kako se paketi nabirajo v strežni vrsti na vozlišču 2 modela MPLS (slika 36), kjer se promet iz generatorja 1 in generatorja 2 združuje. Do časa šestih sekund se paketi nabirajo v zaporedju Z Z R. Pri tem R pomeni paket iz generatorja 1, Z pa paket iz generatorja 2. Po času šestih sekund se vrstni red paketov spremeni, in sicer se paketi nabirajo v zaporedju R Z R. Ker se v strežni vrsti nahaja samo en paket iz generatorja 2, se zakasnitev paketov zmanjša.

Vsaka sprememba zakasnite se odraža na vrednosti jittra, ki ga prikazuje graf b) na sliki 38. Jitter se v modelih IP in MPLS pojavi v času dveh sekund, ko se začne generirati omrežni promet iz generatorja 1. V modelu IP je vrednost jittra med časom dveh in petih sekund konstantna. Zaradi točke zasičenja se v času petih sekund vrednost jittra poveča in začne variirati. Vrednost jittra v modelu MPLS konstantno niha, dokler ne pride do točke, kjer se medprihodni čas paketov v strežni vrsti spremeni. V tej točki se vrednost jittra zmanjša in začne konstantno variirati v majhnih mejah.

Iz grafa c) na sliki 38 je razvidno, da je prepustnost v modelu MPLS skozi čas simulacije konstanta. V modelu IP se prepustnost paketov začne zmanjševati, ko se v času dveh sekund iz generatorja 1 generira omrežni promet. Ko delujeta generator 1 in generator 2, je intenzivnost pošiljanja prevelika, zato pride do točke zasičenja na vozlišču 3 v modelu IP (slika 35) in do izgube paketov. V modelu IP se paketi izgubljajo skozi celotni simulacijski čas, zato prepustnost postopno pada. Prepustnost paketov iz generatorja 2 je odvisna tudi od zakasnitve paketov. Bolj so paketi zakasnjeni, manjša je prepustnost. V modelu IP je delež vseh izgubljenih paketov približno 6-odstoten, v modelu MPLS pa ne opažamo izgub.

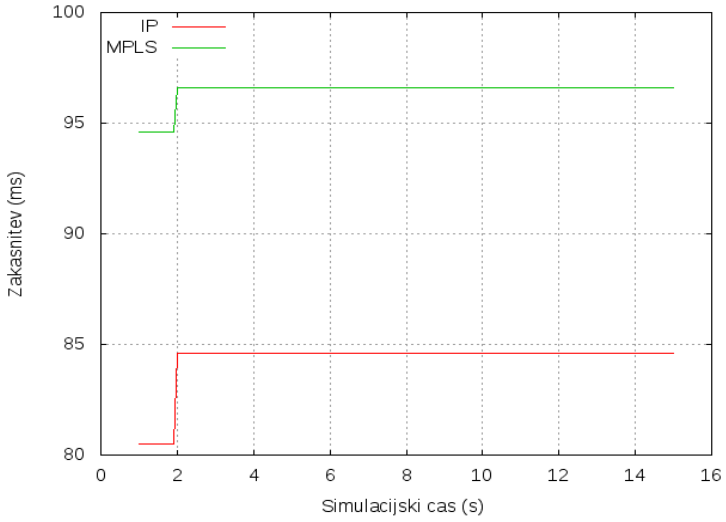
Scenarij 2

V scenariju 2 smo generirali omrežni promet s parametri iz tabele 3 in naslednjimi intenzivnostmi za generator 1 in generator 2:

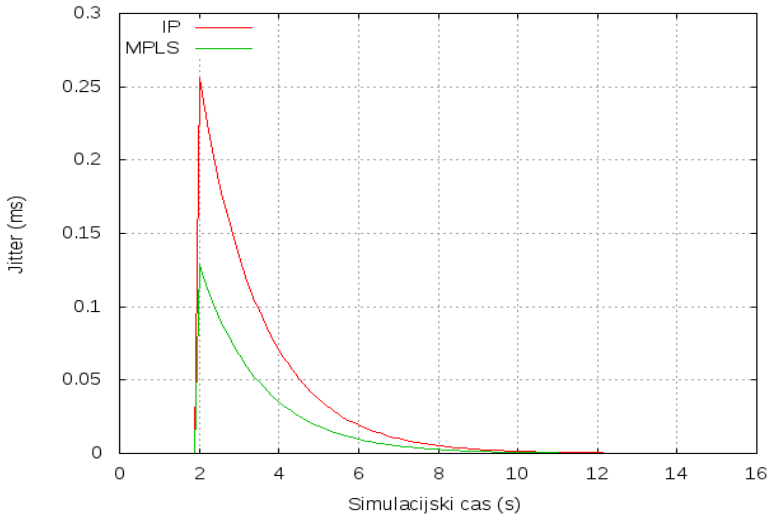
- generator 1: 100 paketov/s,
- generator 2: 10 paketov/s.

Za scenarij 2 pri eksperimentu 1 smo skozi čas simulacije opazovali parametre kakovosti storitev, ki so prikazani na sliki 39.

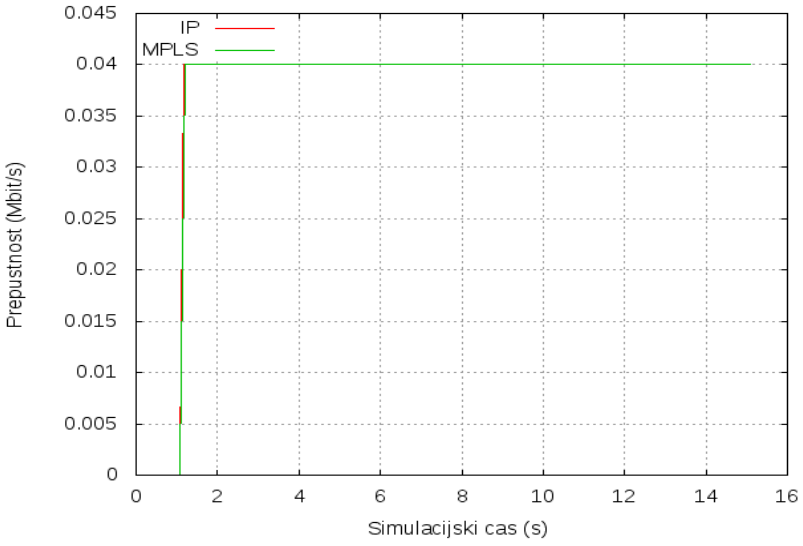
a)



b)



c)



Slika 39: Prikaz parametrov kakovosti storitev za omrežni promet iz generatorja 2 skozi čas simulacije pri scenariju 2 eksperimenta 1 za a) zakasnitev, b) jitter in c) prepustnost

Iz grafa a) na sliki 39 vidimo, da je zakasnitev pred časom dveh sekund manjša v modelu IP kot v modelu MPLS. V času dveh sekund se pojavi povečanje zakasnitve v obeh modelih, zaradi generiranja omrežnega prometa na generatorju 1. Čeprav se zakasnitev poveča, je še vedno manjša v modelu IP kot v modelu MPLS. Skupna intenzivnost generatorja 1 in generatorja 2 je 110 paketov/s in ni prevelika glede na teoretično zgornjo mejo skupne intenzivnosti modela IP, ki je 245 paketov/s. Tako ni potrebe po pošiljanju omrežnega prometa po eksplicitni poti v modelu MPLS.

Eksplicitna pot je daljša, zato se paketi po njej prenašajo z večjo zakasnitvijo. Eksplicitna pot v modelu MPLS vpliva na manjšo spremembo v zakasnitvi, saj se zakasnitev v modelu MPLS poveča za približno 2 ms, v modelu IP pa za približno 4 ms. Večja sprememba zakasnitve v modelu IP je posledica pošiljanja vsega prometa po isti (najkrajši) poti, saj se s tem kopičijo paketi v strežni vrsti vozlišča 2 in vozlišča 3 (slika 35). V modelu MPLS se kopičijo paketi samo v strežni vrsti vozlišča 2 (slika 36), kjer pride do združevanja prometa iz generatorja 1 in generatorja 2.

S spremembo zakasnitve se spremeni tudi vrednost jittra na grafu b) na sliki 39. Vrednost jittra je tako v modelu IP in modelu MPLS izredno majhna v primerjavi z zakasnitvijo obeh modelov. Razmerje med vrednostjo jittra in vrednostjo zakasnitve za model IP je približno 1:350, za model MPLS pa približno 1:800. Jitter se v modelu IP poveča do največje vrednosti 0,25 ms, v modelu MPLS pa največ do vrednosti 0,125 ms. Ko vrednost jittra v modelu IP in modelu MPLS doseže največjo vrednost, se postopno zmanjšuje. Posledica postopnega padanja jittra je iterativnost računanja jittra za vsak paket posebej. Veliko vlogo pri postopnem zmanjševanju jittra ima deljenje z vrednostjo 16 v enačbi 2 (poglavje 8.6). Če nismo uporabili deljenja z vrednostjo 16, smo dobili nagel padec vrednosti jittra.

Ker skupna intenzivnost iz generatorja 1 in generatorja 2 ni prevelika, se ne pojavi izguba paketov za model IP in MPLS. Zaradi tega je v obeh modelih na grafu c) na sliki 39 prepustnost enaka in konstanta skozi čas simulacije.

9.2.2 Eksperiment 2

Pri eksperimentu 2 smo preverili, kako naključno generiranje prometa vpliva na promet z deterministično porazdelitvijo v modelih IP in MPLS. Generator 1 je naključno generiral pakete, katerih verjetnost porajanja je porazdeljena po Poissonovi verjetnosti porazdelitvi. Generator 2 je generiral pakete z deterministično porazdelitvijo v enakih časovnih intervalih. Za generiranje paketov na generatorju 2 se uporabi aplikacija CBR, na generatorju 1 pa aplikacija Poisson. Pošiljanje omrežnega prometa iz obeh generatorjev je potekalo preko transportnega protokola UDP. Generatorja sta generirala pakete velikosti 512B. Za generator 1 in generator 2 so nastavljeni parametri prikazani v tabeli 6.

Za model IP je teoretična zgornja meja skupne intenzivnosti iz generatorja 1 in generatorja 2 približno 245 paketov/s in se izračuna po enakem postopku kot v eksperimentu 1 (enačba 6). Za model MPLS je teoretična zgornja meja približno 490 paketov/s.

	Generator 1	Generator 2
Velikost paketa	512B	512B
Generiranje paketov	naključno	deterministično
Aplikacija	Poisson	CBR
Transportni protokol	UDP	UDP

Tabela 6: Parametri za eksperiment 2

Za generatorje omrežnega prometa veljajo enake intenzivnosti kot pri eksperimentu 1, kjer generator 1 pošilja pakete s fiksno intenzivnostjo 100 paketov/s, medtem ko za generator 2 določimo različne intenzivnosti pošiljanja paketov (10, 50, 100, 160…). Pri različnih intenzivnostih generatorja 2 in fiksni intenzivnosti generatorja 1 smo izmerili povprečne vrednosti zakasnitev (tabela 7) in jittra (tabela 8) v modelih IP in MPLS.

Generator 2 (paketi/s)	IP (ms)	MPLS (ms)
10	81,06	94,91
50	81,52	94,89
100	83,46	94,88
160	**234,61**	**94,84**
200	252,11	94,92
250	385,77	237,67

Tabela 7: Povprečna zakasnitev za eksperiment 2

Če primerjamo povprečne vrednosti zakasnitev modelov IP in MPLS v eksperimentu 1 (tabela 4) s povprečnimi vrednostmi zakasnitev iz tabele 7, ugotovimo, da se povprečne vrednosti zakasnitev kljub naključnem generiranju prometa bistveno ne spreminjajo.

Enako kot v eksperimentu 1 smo na podlagi izmerjene povprečne zakasnitve v tabeli 7 določili intenzivnost za generator 2, s katero smo simulirali parametre kakovosti storitev za modela IP in MPLS. Iz tabele 7 vidimo, da se enako kot pri eksperimentu 1 povprečna zakasnitev v modelu IP in MPLS do intenzivnosti 160 paketov/s ne spreminja. Za povečanje zakasnitve v modelu IP je vzrok enak kot pri eksperimentu 1, in sicer to, da skupna intenzivnost iz generatorja 1 in generatorja 2 preseže teoretično zgornjo mejo skupne intenzivnosti. V modelu MPLS se zakasnitev poveča pri intenzivnosti 250 paketov/s, saj se takrat pojavi točka zasičenja na generatorju.

Generator 2 (paketi/s)	IP (ms)	MPLS (ms)
10	0,79	0,61
50	1,23	0,57
100	1,85	0,49
160	2,11	0,47
200	1,54	0,44
250	1,35	0,13

Tabela 8: Povprečen jitter za eksperiment 2

Povprečne vrednosti jittra (tabela 8) so tudi pri generiranju naključnega omrežnega prometa na generatorju 1 v modelu MPLS vedno manjše kot v modelu IP, ne glede na intenzivnost pošiljanja. Če primerjamo vrednosti jittra iz tabele 5 eksperimenta 1 z vrednostmi jittra iz tabele 8, ugotovimo, da so vrednosti jittra v tabeli 8 tako za model IP kot za model MPLS vedno večje od vrednosti jittra v eksperimentu 1, kar je posledica naključno generiranega prometa na generatorju 1. Zanimiv pojav, do katerega pride v modelu MPLS ob večanju intenzivnosti, je postopno zmanjševanje povprečne vrednosti jittra. Vzrok tega je, da se pri večanju intenzivnosti deterministčnega prometa iz generatorja 2 pojavi v strežni vrsti na vozlišču 2 (slika 36) več deterministčnega prometa, zato naključno generiran promet iz generatorja 1 nima velikega vpliva na promet iz generatorja 2. Sistem se obnaša deterministično, deluje kot nekakšen "buffer", zato se zakasnitve spreminjajo manj, kar se kaže pri vrednosti jittra.

V nadaljevanju nas je zanimalo, kako generator 2 z intenzivnostjo 160 paketov/s in generator 1 z intenzivnostjo 100 paketov/s vplivata na simulacijo parametrov kakovosti storitev v modelih IP in MPLS. To je prikazano v scenariju 1. V scenariju 2 smo opazovali simulacijo parametrov kakovosti storitev, če spremenimo intenzivnost generatorja 1 na 50 paketov/s.

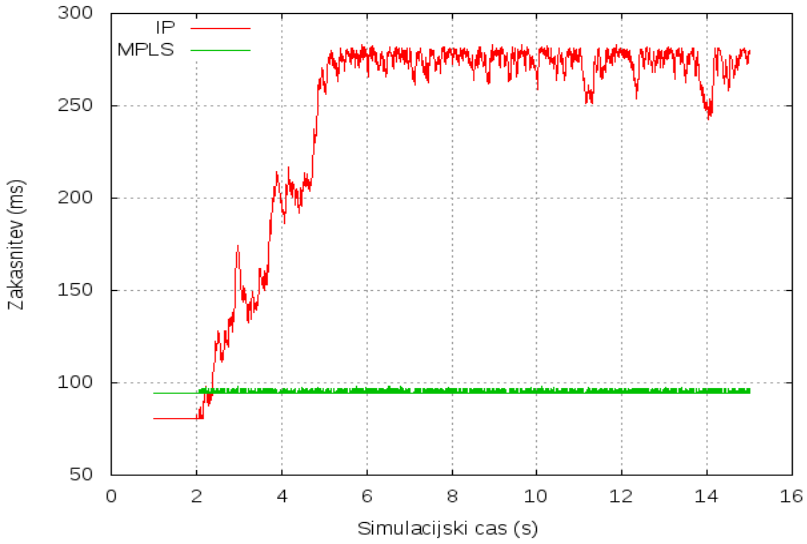
Scenarij 1

V scenariju 1 smo generirali omrežni promet s parametri iz tabele 6 in z naslednjimi intenzivnostmi za generator 1 in generator 2:

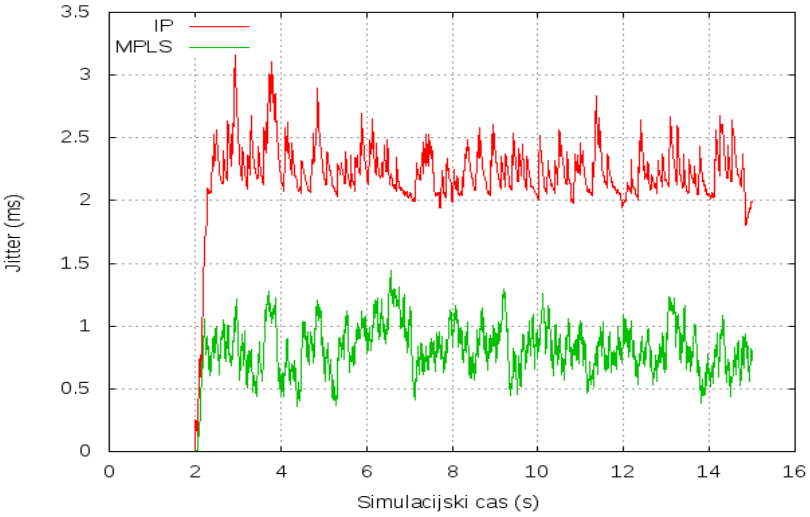
- generator 1: 100paketov/s,
- generator 2: 160paketov/s.

Za scenarij 1 pri eksperimentu 2 smo skozi čas simulacije opazovali parametre kakovosti storitev, ki so prikazani na sliki 40.

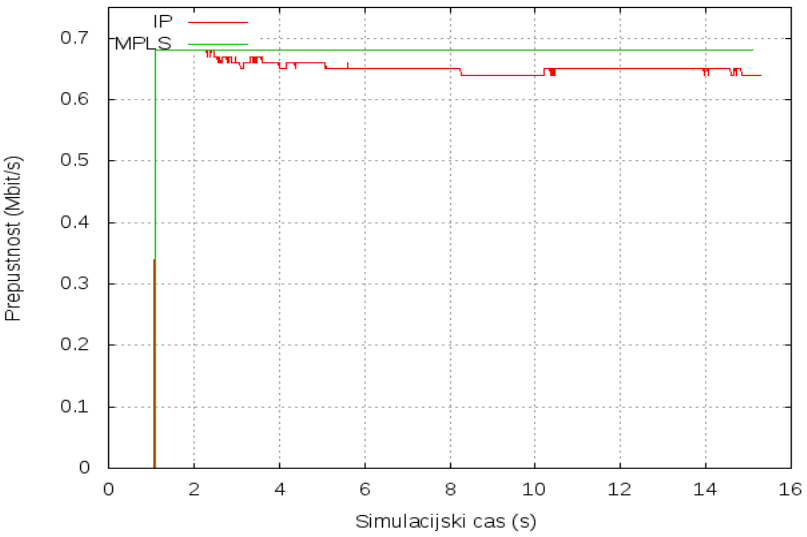
a)



b)



c)



Slika 40: Prikaz parametrov kakovosti storitev za omrežni promet iz generatorja 2 skozi čas simulacije pri scenariju 1 eksperimenta 2 za a) zakasnitev, b) jitter in c) prepustnost

Iz grafa a) na sliki 40 vidimo, da se do časa dveh sekund zakasnitev paketov omrežnega prometa iz generatorja 2 ne spreminja in je v modelu IP manjša kot v modelu MPLS. V času dveh sekund se zakasnitev v modelu IP občutno poveča in preseže zakasnitev iz modela MPLS. Povečanje zakasnitve v modelu IP je posledica naključno generiranega prometa iz generatorja 1. Kljub naključnem generiranju omrežnega prometa iz generatorja 1 se v modelu MPLS ne pojavi opazen porast zakasnitve, temveč se pojavi variiranje zakasnitve v majhnih mejah. Variiranje zakasnitve je v modelu IP bistveno večje, ker se ves omrežni promet iz generatorja 1 in generatorja 2 pošilja po isti (najkrajši) poti.

Spremenljivost zakasnitve vpliva na vrednost jittra, ki ga prikazuje graf b) na sliki 40. V modelih IP in MPLS se jitter pojavi, ko se iz generatorja 1 začne naključno generirati omrežni promet. Skozi čas simulacije je jitter v modelu MPLS manjši kot v modelu IP, saj se omrežni promet iz generatorja 2 pošilja ločeno po eksplicitni poti. Čeprav je vrednost jittra skozi čas simulacije v modelu MPLS manjša kot v modelu IP, je nihanje v modelu MPLS še vedno veliko in je primerljivo z nihanjem v modelu IP. Ugotovimo, da kljub vzpostavitvi eksplicitne poti v modelu MPLS ne moremo popolnoma odpraviti velikega nihanja. Tako ima naključno generiran promet velik vpliv na nihanje vrednosti jittra. Posledica nihanja v modelih IP in MPLS je združevanje prometa iz generatorja 1 in generatorja 2 na vozlišču 2 (sliki 35 in 36).

Iz grafa c) na sliki 40 je prepustnost v modelu MPLS skozi čas simulacije konstanta. V modelu IP se prepustnost začne zmanjševati, ko delujeta oba generatorja prometa. Pri tem se pojavi izguba paketov. Delež vseh izgubljenih paketov je približno 4-odstoten.

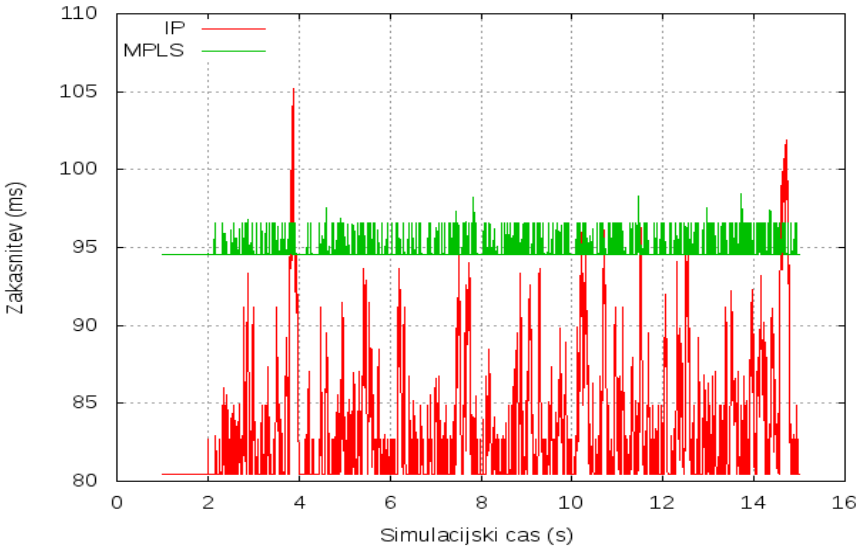
Scenarij 2

V scenariju 2 smo generirali omrežni promet s parametri iz tabele 6 in z naslednjimi intenzivnostmi za generator 1 in generator 2:

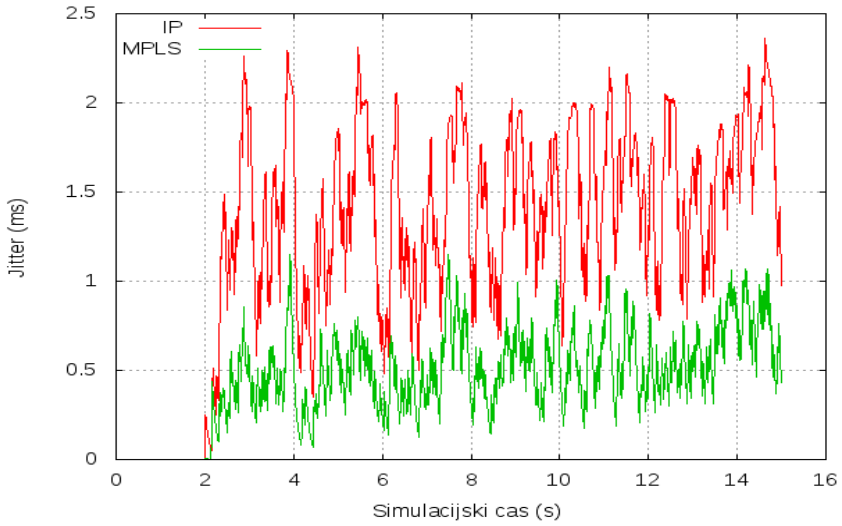
- generator 1: 50 paketov/s,
- generator 2: 160 paketov/s.

Za scenarij 1 pri eksperimentu 2 smo skozi čas simulacije opazovali parametre kakovosti storitev, ki so prikazani na sliki 41.

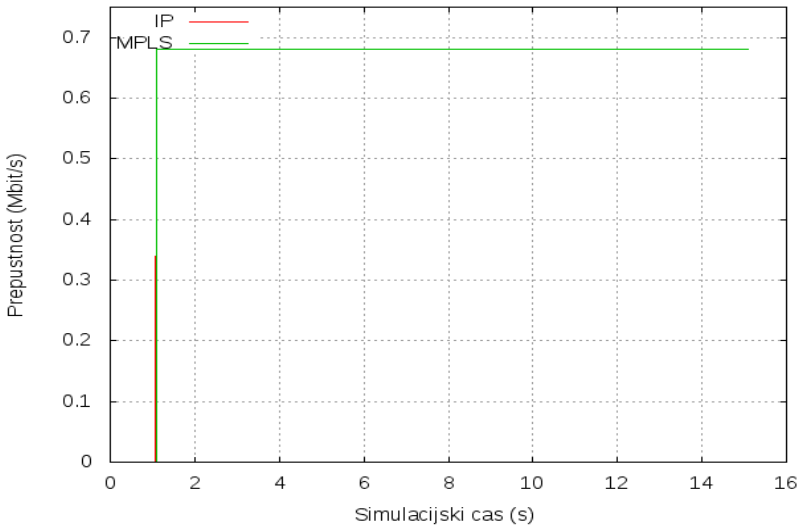
a)



b)



c)



Slika 41: Prikaz parametrov kakovosti storitev za omrežni promet iz generatorja 2 skozi čas simulacije pri scenariju 2 eksperimenta 2 za a) zakasnitev, b) jitter in c) prepustnost

Zakasnitev na grafu a) na sliki 41 je do časa dveh sekund v modelu IP manjša kot v modelu MPLS. V času dveh sekund začne zakasnitev v obeh modelih variirati zaradi generiranja naključnega omrežnega prometa na generatorju 1. Zakasnitev v modelu MPLS variira v manjših mejah kot v modelu IP, ker se v strežni vrsti vozlišča 3 (slika 36) nahaja manj paketov.

Variiranje zakasnitve vpliva tudi na vrednost jittra na grafu b) na sliki 41, kjer jitter v modelu IP variira v večjih mejah kot v modelu MPLS. Skozi čas simulacije je vrednost jittra v modelu MPLS kljub variiranju manjša od vrednosti jittra v modelu IP. Čeprav zmanjšamo intenzivnost naključno generiranega omrežnega prometa na generatorju 1, je nihanje vrednosti jittra v modelu IP in modelu MPLS še vedno zelo veliko.

Ker intenzivnost pošiljanja iz generatorja 1 in generatorja 2 ni prevelika, ne opazimo izgube paketov za modela IP in MPLS. Zaradi tega je v obeh modelih na grafu c) na sliki 41 prepustnost enaka in konstanta skozi čas simulacije.

9.2.3 Eksperiment 3

Pri eksperimentu 3 se v modelih IP in MPLS omrežni promet na generatorju 1 generira z aplikacijo FTP, na generatorju 2 pa z aplikacijo CBR. Z aplikacijo CBR se paketi z velikostjo 64B pošiljajo v konstantnih časovnih intervalih preko protokola UDP. Aplikacija FTP generira 512B velike pakete, ki se pošiljajo preko protokola TCP.

Prenos paketov preko protokola TCP se začne po vzpostavitvi povezave s trismernim rokovanjem (ang. three-way handshake). Sprejemnik za vsak prejeti paket pošlje potrditev v obliki paketa ACK. Ko pošiljatelj prejme paket ACK, nadaljuje s pošiljanjem paketov. Največje število paketov, ki jih pošiljatelj lahko naenkrat pošlje brez čakanja na potrditev paketa ACK, je odvisno od velikosti širine okna, ki ga predhodno nastavimo.

V tabeli 9 so prikazani parametri za generator 1 in generator 2, ki ostajajo skozi eksperiment nespremenjeni. Intenzivnost pošiljanja paketov za generator 1 smo nastavljali z velikostjo okna. Generator 2 je pošiljal pakete s fiksno intenzivnostjo.

	Generator 1	Generator 2
Velikost paketa	512B	64B
Generiranje paketov	odvisno od potrditvene zahteve	deterministično
Aplikacija	FTP	CBR
Transportni protokol	TCP	UDP

Tabela 9: Parametri za eksperiment 3

V eksperimentu 3 smo izvajali simulacijo parametrov kakovosti storitev za modela IP in MPLS z dvema scenarijema. V scenariju 1 smo za generator 1 določili velikost okna na vrednost 40 paketov, za velikost okna v scenariju 2 pa vrednost 80 paketov. V obeh scenarijih se na generatorju 2 generira omrežni promet z intenzivnostjo 125 paketov/s.

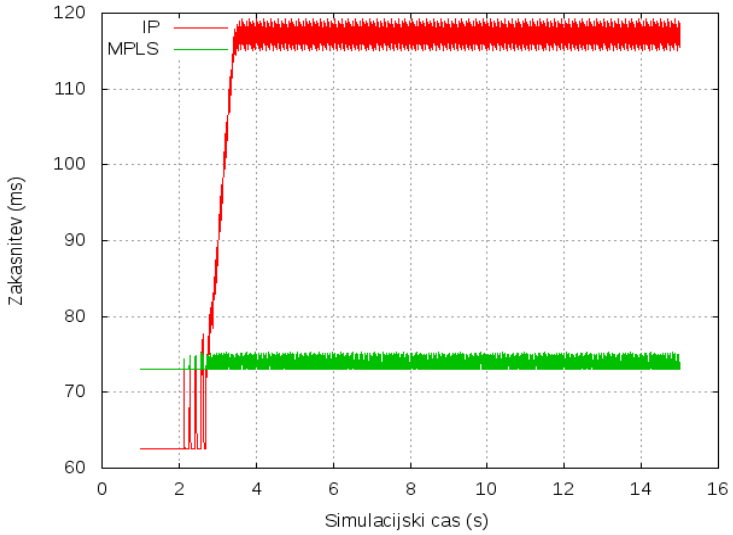
Scenarij 1

V scenariju 1 smo generirali omrežni promet s parametri, prikazanimi v tabeli 9, z vrednostmi za širino okna generatorja 1 in intenzivnostjo generatorja 2:

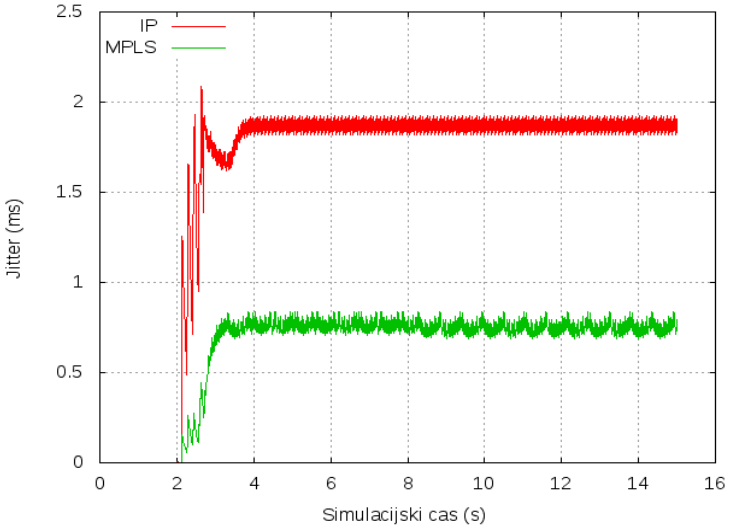
- generator 1 (širina okna): 40 paketov,
- generator 2: 125 paketov/s.

Za scenarij 1 pri eksperimentu 3 smo skozi čas simulacije opazovali parametre kakovosti storitev, ki so prikazani na sliki 42.

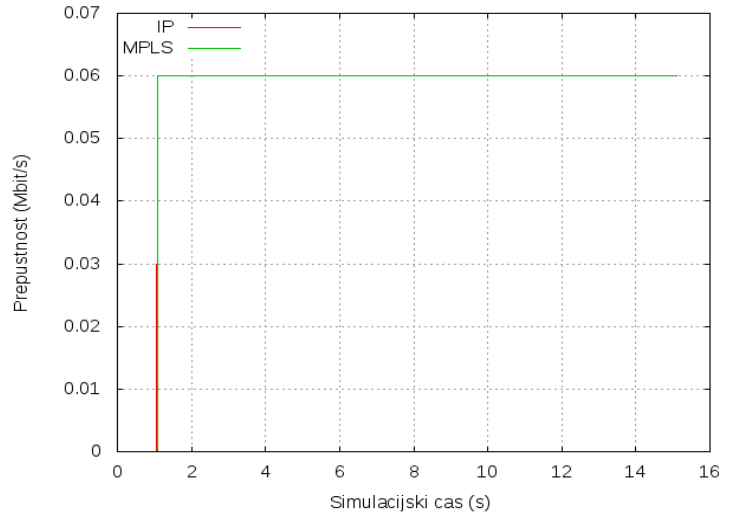
a)



b)



c)



Slika 42: Prikaz parametrov kakovosti storitev za omrežni promet iz generatorja 2 skozi čas simulacije pri scenariju 1 eksperimenta 3 za a) zakasnitev, b) jitter in c) prepustnost

Enako kot v eksperimentu 1 in eksperimentu 2 se do časa dveh sekund pošiljajo paketi v modelu IP z manjšo zakasnitvijo kot paketi v modelu MPLS. To je prikazano na grafu a) na sliki 42. V času dveh sekund se začnejo pošiljati paketi omrežnega prometa iz generatorja 1, ki vplivajo na zakasnitev paketov iz generatorja 2. Zakasnitev se v modelu IP povečuje med časom dveh in štirih sekund, kar je posledica enakomernega povečevanja števila zaporedno poslanih paketov iz generatorja 1. Največje število zaporedno poslanih paketov, ki jih generator 1 lahko naenkrat generira brez čakanja na potrditev paketa ACK, določa širina okna. Ko generator 1 doseže največje število zaporedno poslanih paketov, začne zakasnitev paketov iz generatorja 2 v modelu IP variirati v majhnih mejah. V modelu IP se ne pojavi točka zasičenja, zato je variiranje v majhnih mejah posledica enakomernega zadrževanja paketov v strežni vrsti vozlišča 3 (slika 35). V modelu MPLS je sprememba zakasnitve manjša kot v modelu IP in se enakomerno spreminja skozi čas simulacije, ker omrežni promet iz generatorja 1 ne vpliva na omrežni promet iz generatorja 2.

V modelu IP se vrednost jittra na grafu b) na sliki 42 najprej povečuje približno do vrednosti 2,2 ms in se nato zniža do približne vrednosti 1,6 ms, kjer variira do konca izvajanja simulacije. Vrednost jittra se zniža zaradi spremembe medprihodnega časa paketov (generator 2) v strežno vrsto vozlišča 2 (slika 35). V modelu MPLS jitter enakomerno variira v majhnih mejah skozi celotni čas simulacije.

Ker intenzivnost pošiljanja iz generatorja 1 in generatorja 2 ni prevelika, ne opazimo izgube paketov za modela IP in MPLS. Zaradi tega je v obeh modelih na grafu c) na sliki 42 prepustnost enaka in konstanta skozi čas simulacije.

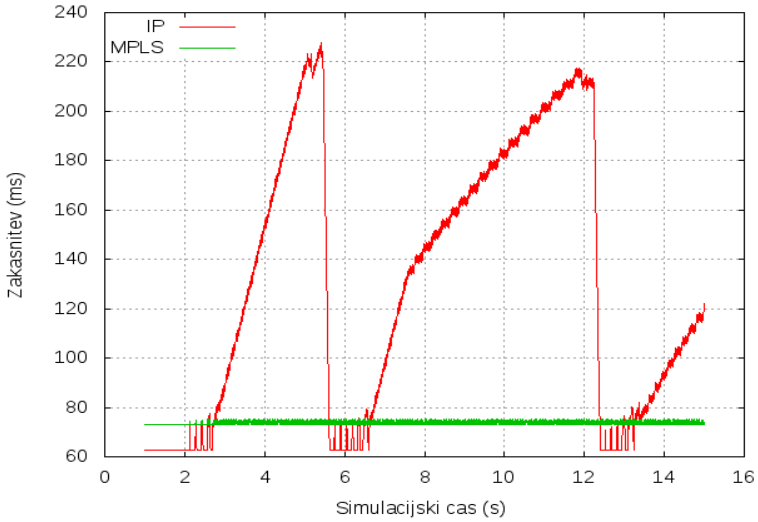
Scenarij 2

V scenariju 2 smo generirali omrežni promet s parametri po tabeli 9 in z naslednjimi vrednostmi za širino okna generatorja 1 in intenzivnostjo generatorja 2:

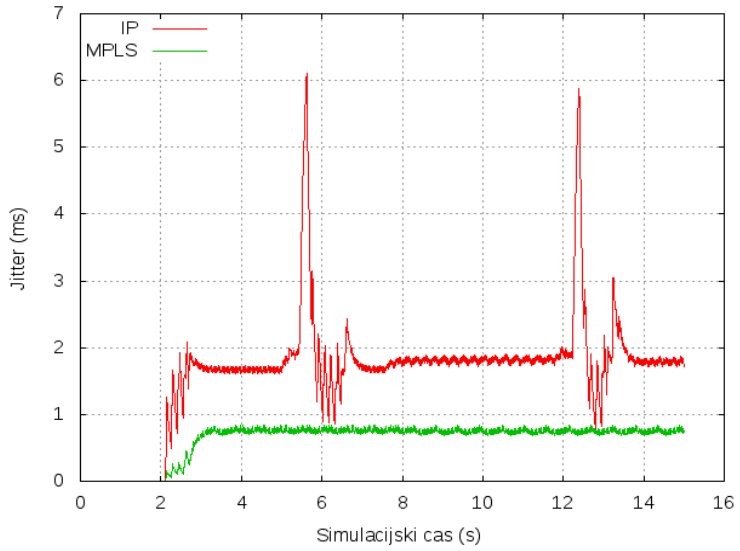
- generator 1 (širina okna): 80 paketov,
- generator 2: 125 paketov/s.

Za scenarij 2 pri eksperimentu 3 smo skozi čas simulacije opazovali parametre kakovosti storitev, ki so prikazani na sliki 43.

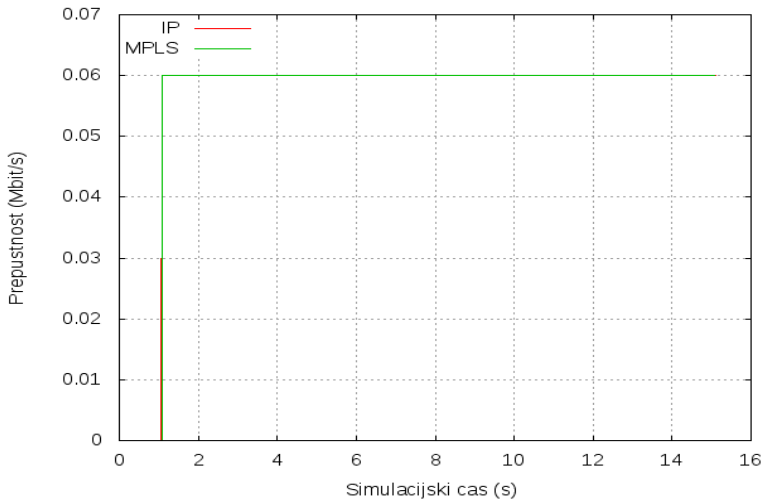
a)



b)



c)



Slika 43: Prikaz parametrov kakovosti storitev za omrežni promet iz generatorja 2 skozi čas simulacije pri scenariju 2 eksperimenta 3 za a) zakasnitev, b) jitter in c) prepustnost

Iz grafa a) na sliki 43 vidimo, da se do časa dveh sekund paketi v modelu IP pošiljajo z manjšo zakasnitvijo kot paketi v modelu MPLS. V času dveh sekund se začnejo pošiljati paketi omrežnega prometa iz generatorja 1, ki vplivajo na spremembo zakasnitve. Zaradi pošiljanja paketov iz generatorja 1 začne zakasnitev paketov iz generatorja 2 v modelu IP linearno rasti, kar je posledica enakomernega povečevanja števila zaporedno poslanih paketov iz generatorja 1. Zakasnitev linearno raste od časa dveh do časa šestih sekund. V tem času se pojavi kopičenje paketov v strežni vrsti vozlišča 3 (slika 35), ker se pošilja preveliko število paketov. Paketi se kopičijo, dokler se strežna vrsta ne zapolni. Zaradi točke zasičenja se začnejo paketi izgubljati. Pri izgubi paketa iz generatorja 1 le-ta preneha pošiljati pakete zato se strežna vrsta na vozlišču 3 sprosti. Pošiljanje se začne ponovno, ko se iz generatorja 1 pošlje zaporedje paketov skupaj s paketi, ki niso bili potrjeni. Zakasnitev paketov iz generatorja 2 zato preneha linearno rasti in pade na vrednost pred linearno rastjo. Tako dobimo skozi čas simulacije v modelu IP zakasnitev paketov v obliki "žage". To lahko vodi do nezaželenih oscilacij v zakasnitvah.

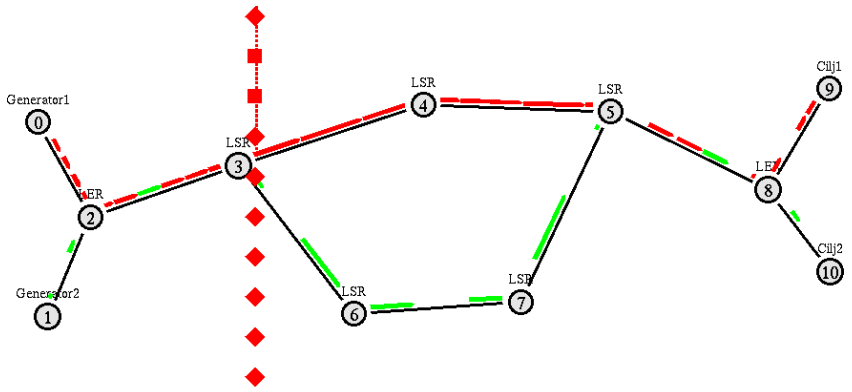
Z uporabo eksplicitne poti v modelu MPLS se zakasnitev paketov iz generatorja 2 enakomerno spreminja skozi čas simulacije. Paketi iz generatorja 1 ne vplivajo na pakete iz generatorja 2, zato se ne pojavijo velika odstopanja v spremembi zakasnitve.

S spremembo zakasnitve se spremeni tudi vrednost jittra na grafu b) na sliki 43. V modelu IP se skozi čas simulacije pojavijo skokovita naraščanja in padanja vrednosti jittra. V modelu MPLS vrednosti jittra enakomerno variirajo, zato ni opaznih odstopanj.

Iz grafa c) na sliki 43 je prepustnost tako v modelu IP kot v modelu MPLS konstanta,saj se paketi iz generatorja 2 ne izgubijo.

9.2.4 Eksperiment 4

Kot smo spoznali pri eksperimentu 1, se zaradi premajhne pasovne širine med generatorjem in vozliščem 2 teoretično ne more pojaviti zasičenje v hrbteničnem omrežju modela MPLS (slika 36). Pri eksperimentu 4 nas zanima, kako vpliva zasičenje v omrežju MPLS na prenos paketov iz generatorja 2. Za potrebe eksperimenta 4 smo prilagodili model MPLS, in sicer tako, da smo povečali pasovno širino med generatorjem in vozliščem 2 ter med vozličem 8 in ciljem na 2 Mbit/s (slika 36). Parametri za generator 1 in generator 2, ki jih uporabljamo pri izvajanju eksperimenta, so enaki kot v eksperimentu 1 (tabela 3). Generator 1 je pošiljal pakete s hitrostjo 1,5 Mbit/s, generator 2 pa je enkrat pošiljal pakete s hitrostjo 0,8 Mbit/s, drugič pa s hitrostjo 0,5 Mbit/s. Ne glede na hitrost pošiljanja paketov iz generatorja 2 se bo zaradi previsoke hitrosti pošiljanja iz generatorja 1 in prenizke pasovne širine povezave med vozliščema 3 in 4 (slika 36) vedno pojavila točka zasičenja na vozlišču 3 (slika 36). S točko zasičenja se pojavi zapolnitev strežne vrste na vozlišču. Pri zapolnjeni strežni vrsti, ki je omejena s strani dolžine strežne vrste, prihaja do izgube paketov.

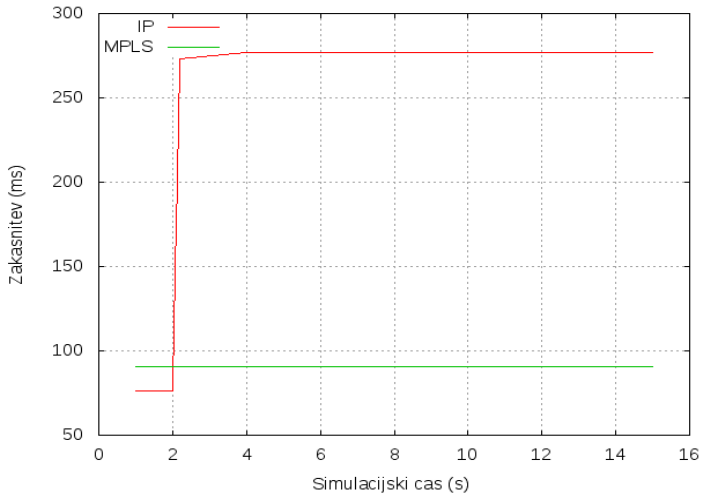


Slika 44:Točka zasičenja na vozlišču 3 v omrežju MPLS

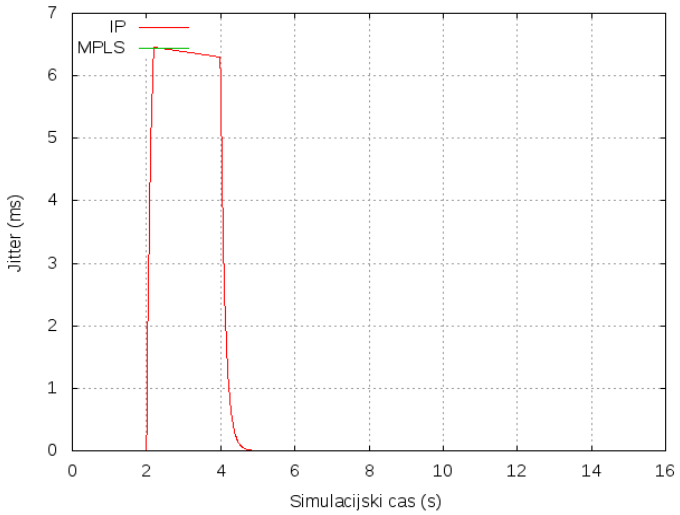
Če generator 1 generira pakete s hitrostjo 1,5 Mbit/s in generator 2 s hitrostjo 0,5 Mbit/s, se pojavi točka zasičenja na vozlišču 3 samo za omrežni promet iz generatorja 1, ker je povezava med vozliščem 3 in 4 ozko grlo omrežja MPLS. Točka zasičenja se pojavi, ko se strežna vrsta na vozlišču 3 (slika 44) zapolni s paketi iz generatorja 1 (rdeča barva). Pri zapolnjeni strežni vrsti so paketi iz generatorja 1 razporejeni v stolpec poleg vozlišča 3, kot na sliki 44. Velikost stolpca predstavlja dolžino strežne vrste, ki je v našem primeru 50 paketov. Če paketi prihajajo v zapolnjeno strežno vrsto, se izgubljajo, kar prikazujejo padajoči rdeči kvadratki na sliki 44. Hitrost pošiljanja paketov iz generatorja 1 je prevelika, zato se paketi nabirajo v strežni vrsti vozlišča 3, dokler se ne začnejo izgubljati. Slednje ne vpliva na pakete iz generatorja 2, ki se prenašajo po povezavi z dovolj veliko pasovno širino. Paketi iz generatorja 2 (zelena barva) se ne nahajajo v strežni vrsti vozlišča 3 (Slika 44), zato se tudi ne izgubijo ob zapolnjeni strežni vrsti. Pri tem se pokaže še ena prednost vzpostavitve eksplicitne poti v omrežju MPLS, ker točka zasičenja na vozlišču 3 ne vpliva na prenos paketov iz generatorja 2, ki se pošiljajo po eksplicitni poti. Parametri kakovosti storitev za omrežni promet iz generatorja 2 se zato ne poslabšajo. V modelu MPLS je za omrežni promet iz generatorja 2 zakasnitev na grafu a) na sliki 45 skozi čas simulacije konstanta, zato se na grafu b) na sliki 45 ne pojavi jitter. Ker ne pride do izgube paketov, se prepustnost na grafu c) na sliki 45 skozi čas simulacije ne spreminja. Tako bi bila eksplicitna pot v omrežju MPLS primerna za pošiljanje omrežnega prometa aplikacij, ki delujejo v realnem času.

V modelu IP, kjer eksplicitna pot ni vzpostavljena, se pojavi izguba zaradi točke zasičenja na vozlišču 3 tako za pakete iz generatorja 1 kot za tiste iz generatorja 2. Izguba paketov iz generatorja 2 je približno 13-odstotna. Za omrežni promet iz generatorja 2 se v modelu IP poslabšajo tudi ostali parametri kakovosti storitev, ker se poveča zakasnitev (graf a) slika 45), pojavi se jitter (graf b) slika 45) in zmanjša se prepustnost (graf c) slika 45). Zaradi delovanja generatorja 1 začne prepustnost omrežnega prometa iz generatorja 2 linearno padati, ker se strežna vrsta vozlišča 3 zasede z omrežnim prometom iz generatorja 1. Prepustnost preneha rasti in začne naraščati, ko se nekaj paketov iz generatorja 1 v strežni vrsti sprosti in pridejo na vrsto paketi iz generatorja 2.

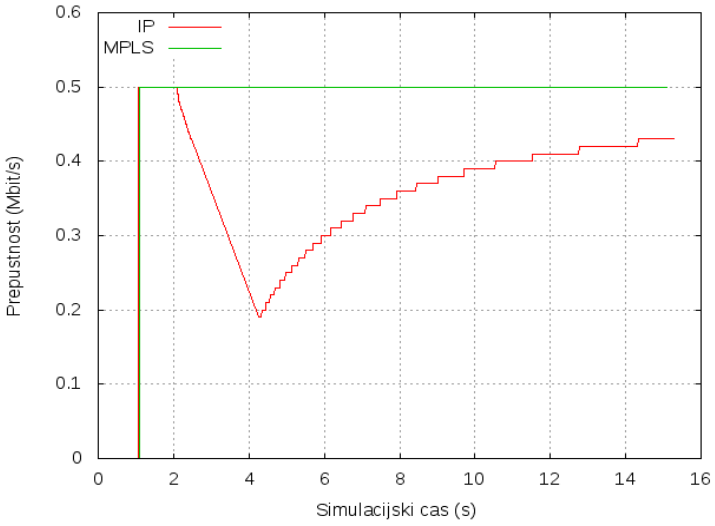
a)



b)

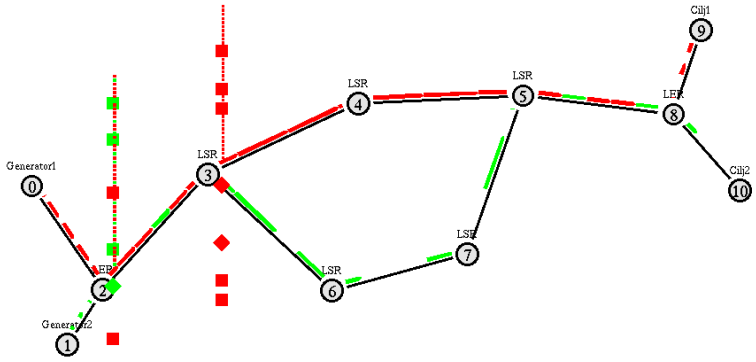


c)



Slika 45: Prikaz parametrov kakovosti storitev omrežnega prometa generatorja 2 skozi čas simulacije pri hitrosti pošiljanja 1,5 Mbit/s generatorja 1 in hitrosti pošiljanja 0,5 Mbit/s generatorja 2 za a) zakasnitev, b) jitter in c) prepustnost

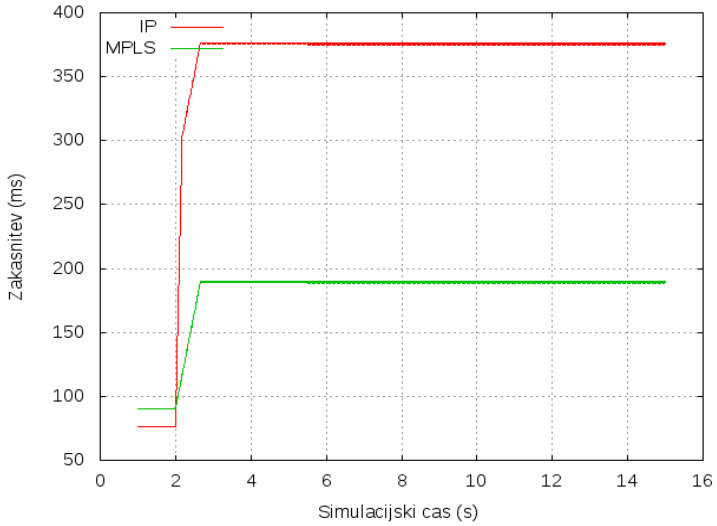
Če generator 2 pošilja pakete s hitrostjo 0,8 Mbit/s, se poleg točke zasičenja na vozlišču 3, ki je prikazana na sliki 46, pojavi tudi točka zasičenja na vozlišču 2. Kot smo že omenili, se točka zasičenja na vozlišču 3 pojavi zaradi ozkega grla med vozliščem 3 in vozliščem 4. Na vozlišču 2 dobimo točko zasičenja zaradi ozkega grla med vozliščem 2 in vozliščem 3 (slika 46), ker je prepustnost povezave 2 Mbit/s in skupna hitrost obeh generatorjev 2,3 Mbit/s. Zaradi prevelike skupne hitrosti se začne polniti strežna vrsta na vozlišču 2 (slika 46) s paketi iz generatorja 1 (rdeča barva) in paketi iz generatorja 2 (zelena barva). Ko se strežna vrsta z velikostjo 50 paketov zapolni, so paketi iz obeh generatorjev razporejeni v stolpec poleg vozlišča 2 na sliki 46. V strežni vrsti oziroma stolpcu vidimo več paketov iz generatorja 1 (rdeča barva), saj je hitrost pošiljanja večja kot pri generatorju 2. Pri zapolnjeni strežni vrsti se začnejo izgubljati tako paketi iz generatorja 1 kot paketi iz generatorja 2. To prikazujejo padajoči rdeči in zeleni kvadratki na sliki 46.



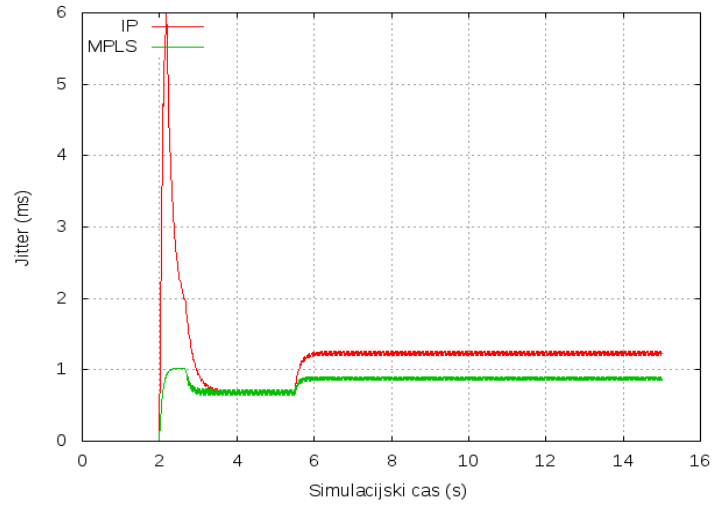
Slika 46: Točka zasičenja na vozlišču 2 in vozlišču 3 v omrežju MPLS

V omrežju MPLS se izguba paketov iz generatorja 2 pojavi zaradi točke zasičenja na vozlišču 2 (slika 46). Izgubljenih paketov iz generatorja 2 je približno 14 %, kar vpliva na zmanjšanje prepustnosti, kot je vidno na grafu c) na sliki 47. Zaradi točke zasičenja se v modelu MPLS poveča zakasnitev, kar prikazuje graf a) na sliki 47. Spremembi zakasnitve sledi sprememba jittra na grafu b) na sliki 47, ki se zaradi različne razporeditve paketov v strežni vrsti skozi čas simulacije spremeni. V primeru modela IP, kjer se ves omrežni promet pošilja po isti poti, se enako kot na vozlišču 2 (slika 46) pojavi točka zasičenja tudi na vozlišču 3. Na vozlišču 3 se poleg paketov iz generatorja 1 izgubljajo tudi paketi iz generatorja 2. Izguba paketov iz generatorja 2 je približno 40-odstotna, zato je prepustnost na grafu c) na sliki 47 bistveno manjša kot pri modelu MPLS. Zaradi dveh točk zasičenja se v modelu IP pojavita večja zakasnitev (graf a) na sliki 47) in večji jitter (graf b) na sliki 47) kot v modelu MPLS.

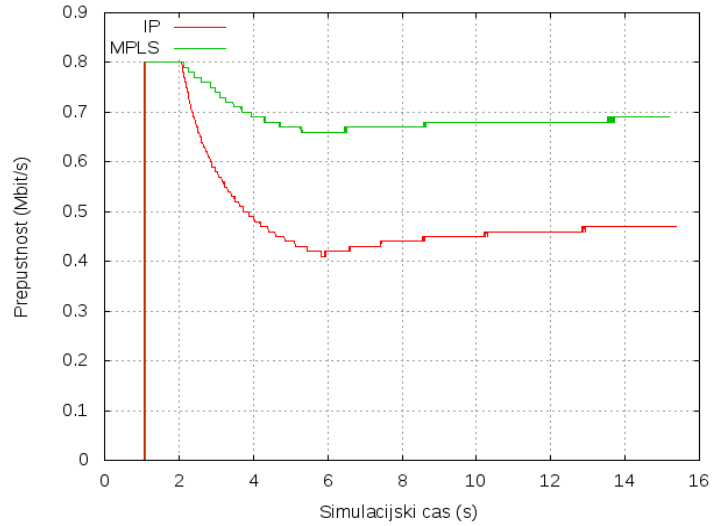
a)



b)



c)



Slika 47: Prikaz parametrov kakovosti storitev omrežnega prometa generatorja 2 skozi čas simulacije pri hitrosti pošiljanja 1,5 Mbit/s generatorja 1 in hitrosti pošiljanja 0,8 Mbit/s generatorja 2 za a) zakasnitev, b) jitter in c) prepustnost

10 Zaključek

V diplomski nalogi smo v sklopu različnih eksperimentov s simulacijo omrežja IP in omrežja MPLS preverili vpliv protokola MPLS na delovanje omrežja IP. Simulacija in implementacija simulacijskih modelov sta potekali s pomočjo odprtokodnega omrežnega simulatorja NS-2. Omrežni simulator NS-2 se je pri analizi rezultatov izkazal za izredno kompleksno orodje, ker ne omogoča samodejne izdelave statistike rezultatov. Rezultate smo dobili s pomočjo programa v programskem jeziku AWK, s katerim smo iz množice podatkov izluščili le tiste, ki so nas zanimali. Program je moral delovati pravilno, saj bi bila sicer analiza rezultatov napačna. Alternativno orodje, ki bi ga lahko uporabili za izvajanje simulacije, bi bil OPNET, ker podpira množico funkcionalnosti, ki v omrežnem simulatorju NS-2 niso na voljo. Ena izmed teh je simulacija aplikacije VoIP.

S simulacijo omrežja IP in omrežja MPLS ugotovimo, da protokol MPLS ne vpliva na hitrost prenosa paketov, kar je bilo v nasprotju z našimi pričakovanji. Pričakovali smo, da bomo z obremenitvijo omrežja MPLS dobili hitrejši prenos paketov kot v omrežju IP. Hitrejši prenos paketov je tako mogoč samo na realno postavljenem omrežju MPLS. Pri izvajanju eksperimentov smo največ pozornosti namenili simulaciji prometnega inženiringa, ki ga omogoča protokol MPLS z vzpostavitvijo eksplicitne poti skozi omrežje MPLS. S simulacijo prometnega inženiringa smo v omrežju IP in v omrežju MPLS spremljali parametre kakovosti storitev. Ugotovili smo, da se pri nizkih obremenitvah omrežja promet v omrežju IP pošilja z manjšimi zakasnitvami kot promet v omrežju MPLS, zato uporaba eksplicitne poti v tem primeru ni smiselna. Upravičenost vzpostavitve eksplicitne poti v omrežju MPLS se izkaže šele pri velikih obremenitvah omrežja. Pri velikih obremenitvah se lahko v omrežju IP pojavi točka zasičenja, kar privede do izgube paketov in do zmanjšanja prepustnosti omrežnega prometa. Temu sledi tudi povečanje zakasnitve in vrednosti jittra za omrežni promet.

Med simulacijami prometnega inženiringa je prišlo do zanimivih pojavov, ki smo jih ustrezno ovrednotili. Najbolj zanimiv pojav je krajšanje medprihodnega časa paketov v strežno vrsto, ki je posledica izbire intenzivnosti pošiljanja omrežnega prometa. Pojav postopnega padanja vrednosti jittra v grafih pripišemo iterativnosti računanja jittra. Zanimiv pojav dobimo pri generiranju determinističnega prometa skupaj z naključnim generiranjem prometa. Z večanjem intenzivnosti generiranja determinističnega prometa se zmanjšuje vpliv naključno generiranega prometa, kar se odraža na postopnem padanju povprečne vrednosti jittra. Pri simulaciji omrežja MPLS, kjer omrežni promet potuje po dveh ločenih poteh, smo dobili zanimiv pojav v primeru ozkega grla na eni izmed poti. Če se na eni izmed poti pojavi ozko grlo in s tem točka zasičenja, to ne vpliva na omrežni promet, ki potuje po drugi poti.

Simulacijski modeli omrežja IP in omrežja MPLS niso predstavljali popolnega stanja realnega omrežja, zato rezultati simulacij podajajo le dovolj dobre približke rezultatov, ki bi jih dobili na realnem omrežju. Po temeljiti analizi rezultatov, ki smo jih dobili pri številnih simulacijah, lahko z gotovostjo trdimo, da smo dobili pričakovane rezultate in zadovoljive ocene za parametre kakovosti storitev pri različnih obremenitvah omrežja IP in omrežja MPLS. Z rezultati simulacij prometnega inženiringa smo dokazali, da je vzpostavitev eksplicitne poti, ki jo omogoča protokol MPLS, pomembna rešitev za optimizacijo omrežja IP. Uporaba eksplicitne poti zagotovi večjo kakovost storitev za aplikacije, ki delujejo v realnem času. Vsekakor pa se ni priporočljivo zanašati samo na simulacijske rezultate, zato je pomembno, da simulacijske modele preizkusimo še s pomočjo meritev na realno postavljenem omrežju. Tako bi bilo zanimivo primerjati rezultate, dobljene na realnem omrežju, z rezultati simulacij.

Dodatek A: Seznam slik

Dodatek B: Seznam tabel

Literatura in viri

[1] A. Khodaskar, S. Ladhake, Multiprotocol Label Switching Protocol, Amravati, Maharashtra, Indija, marec 2011.

[2] A. Kos, Paketna omrežja in kakovost storitev, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2004.

[3] A. Kos, Prenos podatkov v realnem času in zagotavljanje kakovosti storitev v IP omrežjih, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2001.

[4] A. Kos, Prometni inženiring v omrežjih MPLS, Laboratorij za telekomunikacije, Fakulteta za elektrotehniko, Univerza v Ljubljani. Dostopno na: http://www.ltfe.org/wp-content/pdf/prometni_inzeniring.pdf

[5] A. Kos, J. Bešter, Evolucija hrbteničnih IP-omrežij v smeri MPLS, Elektrotehniški vestnik, Ljubljana, 2001. Dostopno na: ev.fe.uni-lj.si/4-2001/kos.pdf

[6] A. Kos, J. Bešter, MPLS v omrežjih ATM, Elektrotehniški vestnik, Ljubljana, 2001. Dostopno na: www.ltfe.org/wp-content/pdf/mpls.pdf

[7] A. Kos, J. Sterle, MPLS, Laboratorij za telekomunikacije Fakultete za elektrotehniko, Univerza v Ljubljani. Dostopno na: http://lt.fe.uni-lj.si/gradiva/NVTSS/Jagodic%20-%202007-2008/14-mpls-akos.pdf

[8] A. Kos, R. Verlič, S. Tomažič, Kakovost storitve v paketnih omrežjih, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2004. Dostopno na: ev.fe.uni-lj.si/3-2004/kos.pdf

[9] A. S. Tanenbaum, Computer Network, 4th edition, New Jersey: Upper Saddle Rover 2003.

[10] D. Savic, J. Bešter, Simulacije v paketnih omrežjih z NS2. Dostopno na: http://lt.fe.uni-lj.si/gradiva/NMVTKO/nmvtko_ns2.pdf

[11] E.Strosar, Vohljati et(h)er(net) in preživeti, Monitor, arhiv februar 2007. Dostopno na: http://www.monitor.si/clanek/vohljati-et-h-er-net-in-preziveti/

[12] G. Ahn, W. Chun, Design and Implementation of MPLS Network Simulator Supporting LDP in CR-LDP, 2000.

[13] H. Hodzic, S. Zoric, Traffic Engineering with Constraint Based Routing in MPLS Networks, 50th International Symposium ELMAR-2008, Hrvaška, september 2008.

[14] H. Osterloh, IP Routing Primer Plus, Sams Publishing, 2002.

[15] IEC tutorial: MPLS. Dostopno na: http://www.iec.org/online/tutorials/mpls/

[16] I. Minei, J. Lucek, MPLS-Enabled Applications, Anglija: John Wiley and Sons, Ltd, 2005.

[17] J. Chung, M. Claypool, NS by Example, Worcester Polytechnic Institute (WPI).
Dostopno na: http://nile.wpi.edu/NS/

[18] J. Metzler, MPLS in Private Networks: Is it a good idea?, Juniper Networks. Dostopno na: http://www.juniper.net/solutions/literature/white_papers/mpls_private.pdf

[19] L. D. Ghein, MPLS Fundamentals, Indianapolis, Cisco Press, 2007.

[20] M. Kukar, Storitev navideznih zasebnih LAN omrežij (VPLS), seminarska naloga pri predmetu Porazdeljeni informacijski sistemi in celovitost podatkov, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2006. Dostopno na: http://www.lkn.fe.uni-lj.si/Seminarji/m_kukar.pdf

[21] M. Porwal, A. Yadav, S. Charhate, Multimedia Traffic Analysis of MPLS and Non MPLS Network, International Journal of Computer Science And Applications Vol. 1, No. 2, avgust 2008.

[22] M. Pustišek, M. Papič, Osnove internetnih sistemov, Elektrotehniški vestnik, Ljubljana, 2001.

[23] N. Zimic, M. Mraz, Temelji zmogljivosti računalniških sistemov, Ljubljana: Fakulteta za računalništvo in informatiko, 2006, str. 35, str. 82.

[24] Omrežni sloj.
Dostopno na: http://www.s-sers.mb.edus.si/gradiva/w3/omrezja/36_osi3/osi3a/osi3.pdf

[25] P. Khatri, MPLS Tutorial, Julij, 2009.
Dostopno na: http://www.sanog.org/resources/sanog14/sanog14-paresh-mpls.pdf

[26] Protokol BGP. Dostopno na: http://en.wikipedia.org/wiki/Border_Gateway_Protocol

[27] Protokol OSPF. Dostopno na: http://en.wikipedia.org/wiki/Open_Shortest_Path_First

[28] R. G. Ingalls, Introduction to simulation, Winter Simulation Conference, 2002, pp. 7–16.

[29] R. Jain, The Art of Computer System Performance Analysis, New York: John Wiley & Sons, Inc., 1991, pogl. 1-7.

[30] S. Klampfer, Simulacija računalniških omrežij, diplomska naloga, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 2006

[31] T. Issariyakul, E. Hossain, Introduction to Network Simulator NS2, Springer, 2009, pogl. 1-9, 13, dodatek A.

[32] U.Hacin, Navidezna zasebna omrežja na osnovi protokola MPLS, magistrsko delo, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2010.

[33] V. Toncar, VoIP Basics: About Jitter.
Dostopno na: http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Jitter.html